MacSADIE 1.2
User's Manual

Digital Image Analysis Lab
Department of Electrical and Computer Engineering
University of Arizona, Tucson, Arizona 85721

Permission is hereby granted to reproduce this
material for any non–commercial purpose.

Apple®, the Apple logo®, LaserWriter®, Macintosh®, and MultiFinder® are registered trademarks,
and Finder™ and QuickDraw™ are trademarks of Apple Computer, Inc.
Other product names mentioned in this manual are the trademarks or registered trademarks of their manufacturers.

Contents

Introduction 7
- Hardware Requirements 7
- Software Requirements 8
- Installation 8
- Startup 8
- Session Log 9
- Images and Windows 9
- Grayscale and Color Display 11
- Selection Rectangle 13
- Menus and Dialogs 14
- Progress Indicator and Background Processing 15
- Error Messages 15
- Macros 16

The Menu 17
- About MacSADIE 17

The File Menu 19
- New 19
  - Checkerboard 19
  - Grayscale 20
  - Sinewave 21
  - Function 21
  - Random 23
- Open 24
- Import 24
- Close 26
- Save 27
- Save As 27
- Page Setup 28
- Print 28
- Quit 29

The Edit Menu 31
Undo 31
Cut 31
Copy 31
Paste 31
Clear 32
Select All 32
Show Clipboard / Hide Clipboard 32
Hide Session Log / Show Session Log 32
Preferences 32
The Analysis Menu 33
Image Information 33
Statistics and Histogram 34
Contour Map 35
Covariance 36
Power Spectrum 37
RMS Difference 38
Scattergram 38
The Contrast Menu 41
Normalize Stretch 41
Histogram Equalize Stretch 42
Linear Stretch 43
Logarithmic Stretch 44
Piecewise Linear Stretch 45
Square Root Stretch 46
Square Stretch 46
Table Lookup Stretch 47
Requantize Graylevels 47
Saturate Graylevels 48
Threshold Graylevels 49
Adaptive Stretch 50
Reference Stretch 50

The Filter Menu 53
FFT Convolution 53
FFT Filter 54
FFT 55
FHT 56
Convolution 57
Gradient (Library Mask) 59
Gradient (User Mask) 60
Variance Map 61
Diversity Map 62
Majority Filter 63
Median Filter 64
Minimum Filter 65
Maximum Filter 65
Conditional Average Filter 66
The Geometry Menu 69
Linear Scale 69
Average and Subsample 70
Window and Subsample 71
Horizontal Mosaic 71
Vertical Mosaic 72
General Mosaic 74
Insert 75
Control Points 75
Polynomial Warp 77
Rotate 79
Mirror 79
The Miscellaneous Menu 81
Combine 81
Compare 82
Principal Components Transform 83
RGB → HSV 84

HSV → RGB 85
BIL → BSQ 86
BSQ → BIL 86
Complex → Real 87
Real → Complex 88
Grid 88
The Classify Menu 91
Levelslice 91
Cluster 92
Segment 93
Signature Map 95
The Windows Menu 97
Full Color 97
Clipboard 97
Session Log 97
Window Entries 97
Appendix A: Mathematical Notation 99
Appendix B: Keyboard Shortcuts 100
Menu 100
Dialog 101
Appendix C: MacSADIE Image File Format 102
Appendix D: MacSADIE Problem Report 103
Index 104

Introduction

MacSADIE is a Macintosh application for processing grayscale or color images for scientific and engineering applications, teaching, and research. It contains more than 50 commands for image analysis, contrast enhancement, spatial filtering, geometric processing, multispectral processing and classification. MacSADIE was conceived at the University of Arizona by Dr. Robert Schowengerdt of the Digital Image Analysis Lab, and designed and written by Gerhard Mehldau of the Department of Computer Science, with algorithm contributions from students in the Digital Image Analysis Lab. This manual describes MacSADIE Version 1.2, which is based on SADIE Version 4.2 (System at Arizona for Digital Image Experimentation), a package of portable C functions in the public domain. For more information on SADIE see the SADIE 4.2 User's Manual.

MacSADIE offers a wide variety of choices for analyzing and processing digital images. According to function, they are grouped into the Analysis, Contrast, Filter, Geometry, Miscellaneous, and Classify menus. In addition, the File menu provides for the creation of synthetic images and the input and output of images in different formats, the Edit menu allows for the transfer of data between applications, and the Windows menu controls MacSADIE's color mode and gives easy access to any window on the desktop.

The first chapter of this manual describes the design and operating principles of MacSADIE. It assumes that you are already familiar with the Macintosh environment; in particular, it assumes that you know about windows, menus, icons, and dialogs, and that you know how to point, click, select, and drag. If you do not recognize these terms, you should refer to the Apple manual that accompanies your system software.

The remainder of this manual is organized into chapters by menus. Each chapter contains a short introduction to the image processing functions provided by the "current" menu's commands. Individual commands are then described in separate sections. The first paragraph of each section gives a brief synopsis of the command, its parameters, and their default values. The second paragraph provides more in-depth coverage, such as a mathematical formulation of the particular image processing function (see Appendix A for a summary of the notation used), reference publications, expert advice on when and how to use a particular command, useful tips, and related commands. To simplify references to MacSADIE commands from menus other than the current one, those commands are referred to as Menu/Command.

The manual assumes a knowledge of basic image processing concepts and terminology. For an introductory text on image processing see, for example, R. A. Schowengerdt, *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York, NY, 1983, or J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction*, Springer Verlag, Berlin, Germany, 1986.

Hardware Requirements

MacSADIE requires a Macintosh with at least an MC68020 processor and a floating-point coprocessor (FPU), together with an RGB color monitor and a video card capable of displaying eight (or more) bits of color.

MacSADIE is set up to request 1024 KBytes (1 MByte) of RAM, but we strongly recommend that this number be increased (with the Get Info command from the Finder's File menu), especially if images are to be displayed in Full Color mode.

MacSADIE will run from a floppy disk; however, for performance reasons we recommend installing MacSADIE on a hard disk drive.

Software Requirements

MacSADIE Version 1.2 requires **System Software Version 6.0.5** (or later), together with **32-Bit QuickDraw Version 1.2** (or later). MacSADIE is compatible with Version 6.0 or later of MultiFinder. Some computation-intensive MacSADIE functions are capable of executing in the background under MultiFinder.

Installation

On a Macintosh without a hard disk, simply insert the MacSADIE disk into the floppy drive. On a Macintosh equipped with a hard disk, we recommend creating a separate folder for MacSADIE. This folder can hold the application itself (copied from the original MacSADIE disk), as well as any image files created with MacSADIE.

If you purchased a single user license for MacSADIE, you need to register the software to your Macintosh before it can be used. Simply follow the instructions in the dialog.

If you obtained a temporary license for MacSADIE, a dialog will remind you at startup of the expiration date of the license, and provide information on how to receive a permanent license.

Startup

Before starting MacSADIE, you should verify that your monitor is set to at least 256 colors. You can determine and, if necessary, change the setting with the Monitors cdev (control panel device) under the /Control Panel command.

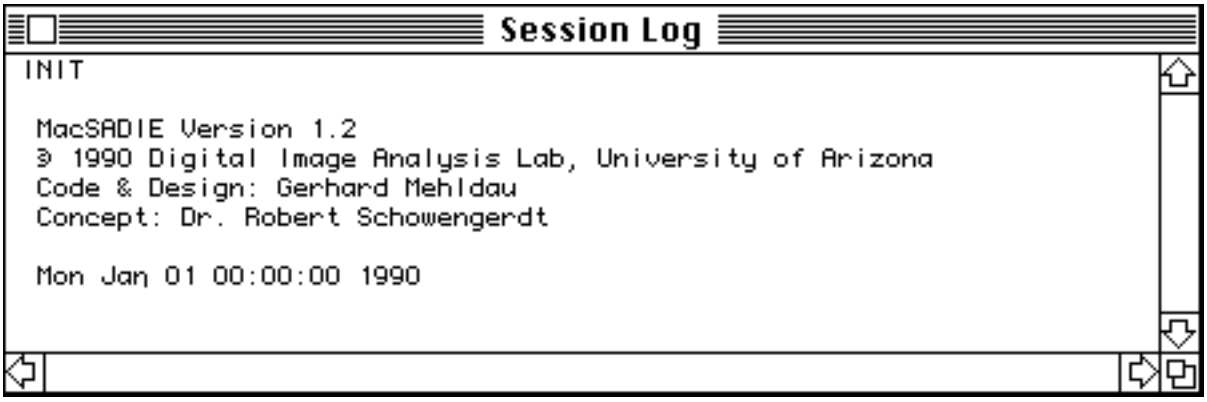
To invoke MacSADIE, simply double-click on the MacSADIE application icon. This is equivalent to selecting the MacSADIE application icon and then choosing the Open command from the Finder's File Menu. Alternately, you can select and open (or double-click on) one or more MacSADIE image icons on the Macintosh desktop. To open PICT or TIFF images directly from the desktop, you need to simultaneously select and open the MacSADIE application together with the image(s) (this requires that the files reside in the same folder as the application).

After startup, MacSADIE installs its own color lookup table (for more information see the section on Grayscale and Color Display below), replaces the previous menu bar with its own, and, if you invoked MacSADIE by opening MacSADIE, PICT, or TIFF images, displays those images on the desktop. It also opens a session log window at the bottom of the main screen. It then waits for you to initiate actions via the mouse or the keyboard.

To print one or more MacSADIE, PICT, or TIFF images directly from the desktop, you can select the image(s) as described above, and then choose the Print command from the Finder's File menu. In this case, MacSADIE will prompt you with the Print command's dialog, print the image(s) as specified, and immediately return to the desktop.

Session Log

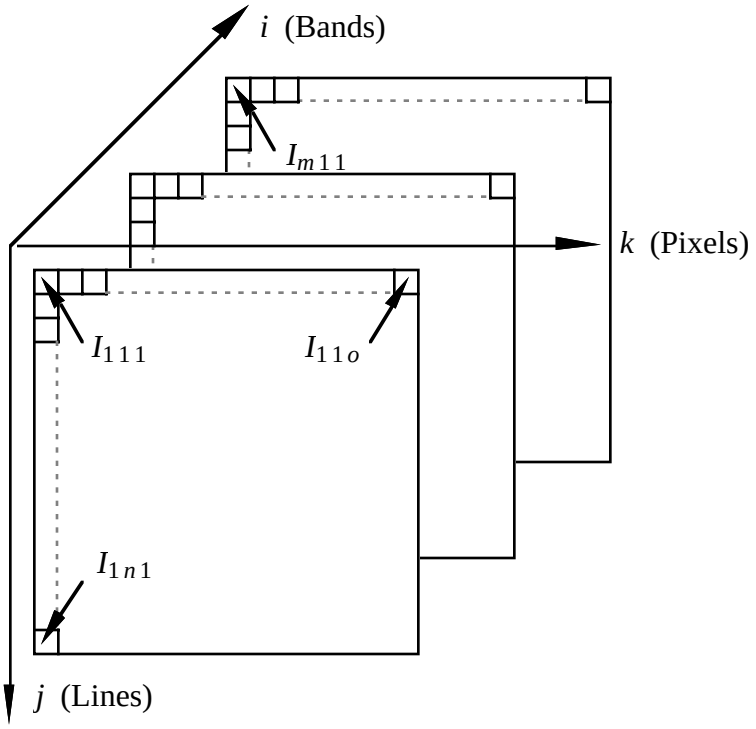
MacSADIE maintains a record of the current session in a window at the bottom of the main screen. This session log contains a list of the SADIE image processing functions, in the order in which they were called from MacSADIE, together with any textual or graphical output from these functions. Examples of textual output are listings of the masks used in the Filter/Convolution command, or the coefficients computed by the Geometry/Control Points command. The graylevel profiles produced by the Analysis/Image Information command or the histograms created by the Analysis/Statistics and Histogram command are examples of graphical output that is kept in the session log.


Session Log
INIT
MacSADIE Version 1.2
© 1990 Digital Image Analysis Lab, University of Arizona
Code & Design: Gerhard Mehldau
Concept: Dr. Robert Schowengerdt
Mon Jan 01 00:00:00 1990

You can resize the session log window and move it around the screen as you like. The Edit/Hide Session Log command lets you hide the session log window entirely (and also make it visible again). If your Macintosh is connected to a printer, you can at any time format and print out the contents of the session log with the File/Page Setup and File/Print commands.

Images and Windows

Digital images in MacSADIE may be thought of as three-dimensional arrays I_{ijk} as shown below. Image coordinates for an *m*-by-*n*-by-*o* image range from one to *m* bands (front-to-back), one to *n* lines (top-to-bottom) and one to *o* pixels (left-to-right), respectively. Each pixel has *m* graylevel values associated with it, one for each band in the image. The full precision of real numbers on the computer is used to represent graylevels during processing, eliminating virtually any concern about unwanted numerical truncation or overflow. Thus, a disk file image with eight bits/pixel will be converted upon opening to a memory-resident image with possible graylevels 1.0, 2.0, …, 255.0. Subsequent image processing may introduce non-integer pixel values, which will be retained during processing. Upon saving a processed image to disk, the default storage format is 32 bits/pixel (i.e., the data is stored in floating point format), but you have the option to pack the data to fewer bits/pixel.



MacSADIE stores all images in memory, with each pixel value represented as a floating point quantity. This approach has two advantages: image data (i.e., individual pixels) can be accessed in random order without speed penalty, and the floating point representation avoids quantization errors that would be introduced by integer arithmetic with a limited number of bits. The drawback, however, is the amount of memory required to store an image: at four bytes/pixel, a 1-by-256-by-256 image requires 256 KBytes of memory. In addition, each image has associated with it a 512 Byte header, some overhead for window management, and an off-screen pixel map to speed up display of the image. Depending on the color mode, the off-screen pixel map requires either one or four bytes per pixel. Color mode is described in detail in the section on Grayscale and Color Display. The memory requirements (in KBytes, without image header and window overhead) for some common image sizes are summarized in the table below.

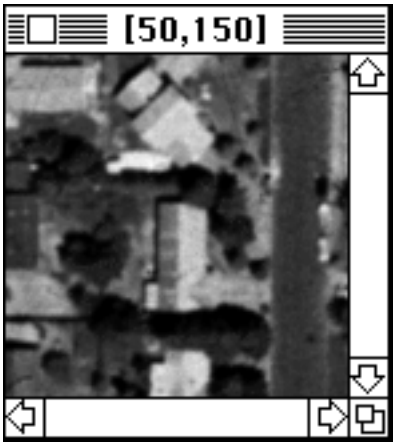
Image Size	Image	Off-screen pixel map Full color off / on	Total Full color off / on
1-by-128-by-128	64	16 / 64	80 / 128
1-by-256-by-256	256	64 / 256	320 / 512
1-by-512-by-512	1024	256 / 1024	1280 / 2048
3-by-128-by-128	192	16 / 64	208 / 256
3-by-256-by-256	768	64 / 256	832 / 1024
3-by-512-by-512	3072	256 / 1024	3328 / 4096

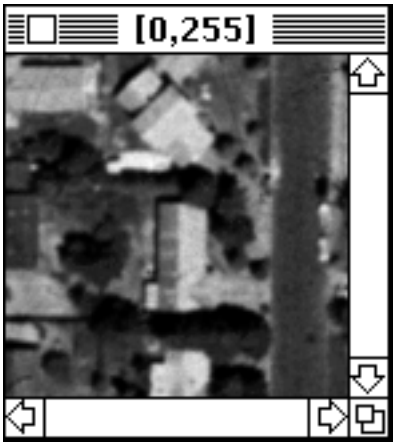
Images are displayed on the desktop in standard Macintosh windows. MacSADIE supports systems with multiple monitors with potentially different screen depths. You can reposition a window by dragging its title bar across the screen, change its size by dragging the size box in the lower right corner, and re-order it with respect to fore- or background. In addition, you can scroll the image if the window is smaller than the image it contains. Clicking on the close box of a window is equivalent to choosing the File/Close command, i.e., the window is removed from the desktop, and all the memory associated with it is returned to the system. You can close all of MacSADIE's windows with a single mouse-click by holding down the Command key ( ) when clicking on the close box of the frontmost window.

Grayscale and Color Display

MacSADIE is capable of simultaneously displaying grayscale and color images on the desktop. One-band images are displayed in grayscale format, three-band images are displayed in color (RGB) format, and all other images have only their first band displayed in grayscale format. This manual refers to those images as "single-band" (grayscale), "three-band" (color), and "multi-band," respectively. Most MacSADIE commands can process all three types of images; in particular, both the File/Open and File/Import commands can read (and, in the case of the File/Import command, simultaneously subsample) images with any number of bands. Commands which only process certain types of images are clearly noted in this manual.

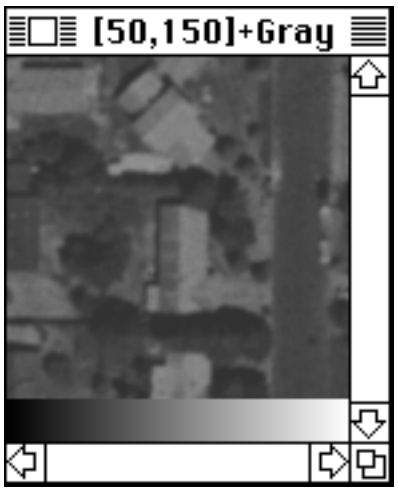
When displaying images, MacSADIE stretches the graylevel range of each image so as to utilize the full range of available grays (or colors) on the display. This is a convenient feature for the user, who doesn't have to be concerned about the graylevel range of an image matching that of the display system, but it also has a non-obvious consequence — different graylevel values in different images may be displayed on the screen with the same intensity value. Consider, for example, an image with a graylevel range from 50.0 to 150.0, and a version of the same image that has been linearly stretched (using Contrast/Linear Stretch) to have a graylevel range from 0.0 to 255.0, as shown below.



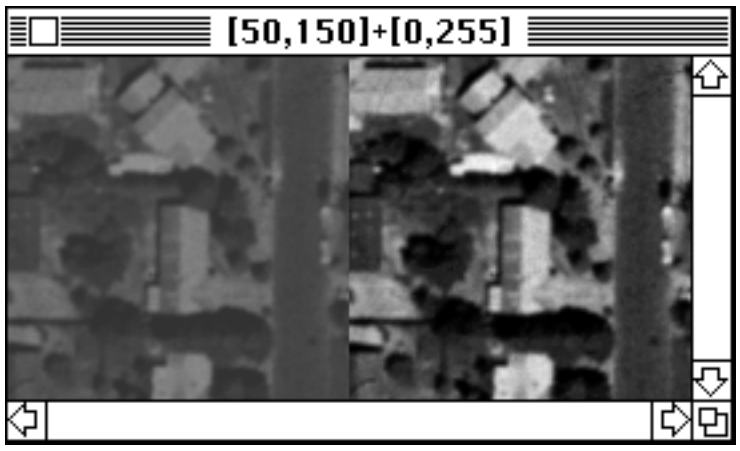


As a result of the automatic display stretch, these two images look identical on the screen. It is important to keep this in mind when attempting to visually compare images on the screen. We can achieve a correct relative comparison of the contrast in the two images above in two ways.

One is to create a grayscale image with a graylevel range from 0.0 to 255.0 and append it to each image (using Geometry/Vertical Mosaic), as shown below.



Now the minimum and maximum graylevels in each modified image are the same (and equal to 0.0 and 255.0, respectively), and they appear on the screen with the correct relative contrast. Another way to compare the two original images is to mosaic them into one image (using, for example, the Geometry/Horizontal Mosaic command), as shown below.



Again, by virtue of combining the two images into one, they are displayed using the same minimum and maximum graylevels (0.0 and 255.0, respectively) derived from the composite image.

The discussion above applies equally well to display and comparison of color images, in which case the three image bands are automatically and individually stretched to the full range of available colors on the display. Note, however, that to append a grayscale image to a color image requires that the grayscale image also have three bands. This can be done by combining three grayscale images with Misc/BSQ → BIL.

Two factors combine to control the appearance of grayscale and color images — the Monitors setting of the screen, and the color mode of an image.

The Monitors setting determines the maximum number of colors that can be displayed simultaneously on the screen. The Monitors setting can be changed with the Monitors control panel device (cdev) under the /Control Panel command. Possible settings range from two colors (i.e., black and white) to 256 colors to "millions of colors," depending on the video card and monitor used. To achieve acceptable display quality, a Monitors setting of at least 256 colors is required. With a Monitors setting of 256 colors, MacSADIE displays color and grayscale images by dividing the 8-bit color lookup table (256 entries) into two halves of 7 bits (128 entries) each. One half holds a 7-bit grayscale, the other half holds a 7-bit color spectrum. MacSADIE creates and installs this custom lookup table at startup (which explains the screen's color change you may notice at this point). With a 24-bit video card and a Monitors setting of "millions of colors," MacSADIE is capable of displaying color and grayscale images in "true color."

The color mode of an image determines the mapping between pixel values and colors. There are two possible modes, which can be set for each image individually with the Windows/Full Color command. In Full Color mode, the red, green, and blue components of a pixel are combined to yield one of 2^{24} colors. Alternately, the color components may be combined to yield either one of 2^7 shades of gray, or one of 2^7 colors (with 2 bits assigned to red, 3 bits to green, and 2 bits to blue). The mapping from pixel values to colors is done with a so-called "off-screen pixel map," which is created together with each image. With Full Color mode turned on at the time of image creation, the pixel map requires four bytes of memory per pixel; with Full Color mode turned off, only one byte per pixel is allocated.

Taken together, the Monitors setting and the color mode result in the following four combinations:

- Monitors setting: 256 colors; color mode: Full Color off. This is the recommended setting, as it results in a good display of grayscale images, and a reasonable display of color images, while minimizing memory requirements and maximizing screen refresh speed.
- Monitors setting: Millions of colors; color mode: Full Color off. This combination is useful in tight memory situations, or for displaying grayscale images — as the difference between a 7-bit and an 8-bit grayscale is virtually imperceptible.
- Monitors setting: 256 colors; color mode: Full Color on. This is the worst possible combination, since it requires large amounts of memory, while not improving the display quality.
- Monitors setting: Millions of colors; color mode: Full Color on. This setting results in the optimum display quality ("true color"), but requires large amounts of memory, a 24-bit video card, and causes a slight reduction in screen refresh speed.

Selection Rectangle

Each MacSADIE image has associated with it a so-called selection rectangle, which determines the image section to be transferred to the clipboard with the Edit/Cut and Edit/Copy commands. The selection rectangle is displayed and can be modified only for the image in the frontmost window; however, it persists when a window is put in the background. Initially, each selection rectangle is empty. To define an arbitrary selection rectangle for the image in the frontmost window, move the cursor to one corner of the rectangle, push (and hold down) the mouse button, drag the cursor to the opposite corner and release the mouse button. The image scrolls automatically if you move the cursor outside the window. You can extend an existing selection rectangle by holding down the Shift key when making the selection. You can also use

the Edit/Select All command to get a selection rectangle that covers the entire image. To clear the selection rectangle, you can either use the Edit/Clear command, or you can double-click anywhere in the image.

Menus and Dialogs

MacSADIE commands are normally selected by clicking on the menu bar, pulling down a menu, dragging the mouse to the desired command, and releasing it there. Some frequently used commands (e.g., File/Open, File/Save, and File/Quit) can also be invoked directly from the keyboard by holding down the Command key ( ) and typing the single-letter command shown on the menu (O, S, and Q, respectively, for the commands in the example above). All menu Command key shortcuts are summarized in Appendix B.

MacSADIE's image processing commands (i.e., commands on menus other than the File, Edit, and Windows menus) operate on the image in the frontmost window, and create a new window containing the processed image. Individual commands are enabled and disabled depending on this image and its characteristics. If, for example, the frontmost image is a single-band (grayscale) image, all commands requiring a three-band (color) image are disabled. The exact conditions under which each command is enabled are described in the sections on the individual menus.

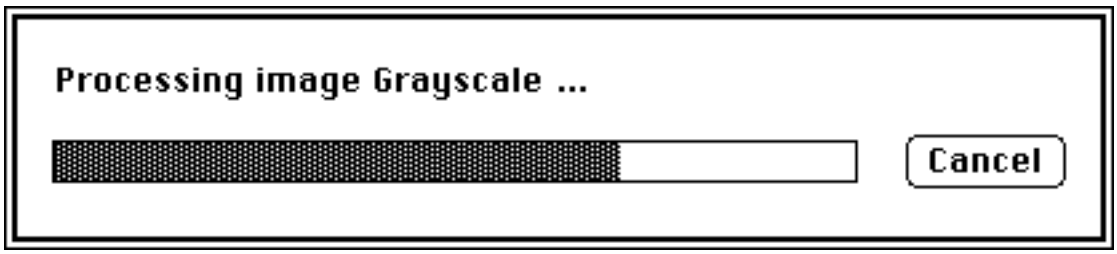
When you select a command from a menu, MacSADIE prompts you for additional parameters to the command with a dialog. Most MacSADIE dialogs have four buttons arranged as shown in the example at the right. Clicking on the OK button initiates processing. Pressing the Return or Enter key is equivalent to clicking the OK button. If you click on the Help button, MacSADIE displays another small dialog containing a concise description of the command's function. Instead of clicking on the Help button, you can also press Command-? or Command-H. The Expert button is currently disabled in all dialogs — it will provide you with more detailed information on the chosen command in a future version of MacSADIE. The Cancel button lets you change your mind and return to the desktop without any processing being done. Pressing Command-. (period) or Command-C is equivalent to clicking on the Cancel button. All dialog Command key shortcuts are summarized in Appendix B.

All commands that lead to the creation of a new image have an "Image Name" parameter in their dialog, to let you specify a name for the new window/image. The default name is an indication of the command used to create the image, but it may be set to anything you desire.

MacSADIE provides reasonable default values for all parameters. Most defaults are static, i.e., as shown in the sample dialog for each command. However, some values are computed dynamically as a function of the characteristics of the frontmost image. If that is the case, the section on the particular command explains exactly how those defaults are computed. If you know in advance that the defaults are acceptable, you can skip over the dialog entirely by holding down the Shift key when selecting the command.

Progress Indicator and Background Processing

Most MacSADIE functions display the wristwatch cursor while processing an image. However, when executing some particularly computation-intensive functions (e.g., the Filter/FFT command), MacSADIE displays a progress indicator as shown below. The Cancel button lets you abort the computation in progress at any time. Pressing Command-. or Command-C is equivalent to clicking on the Cancel button.

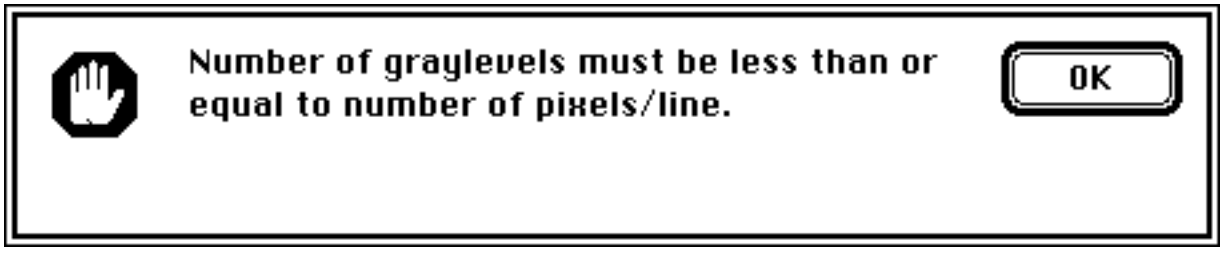


Functions that display this progress indicator are capable of executing in the background. If MacSADIE is running under MultiFinder, you can switch to any other application (including the Finder) by either clicking on a window belonging to that application, or by clicking on the small icon at the right end of the menu bar. Be advised, however, that putting such a computation in the background will increase the amount of time required for the command to complete. The exact amount of the increase depends on the particular foreground application, and the type of activity you are performing with it.

Error Messages

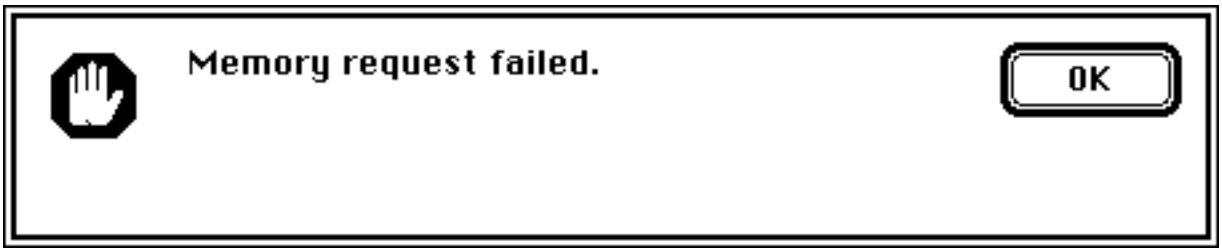
Occasionally, MacSADIE may present you with an error message. All error messages are intended to be self-explanatory. There are two types of errors reported — input errors and memory-related errors.

Input errors are usually caused by inconsistent parameters. MacSADIE commands check all their parameters before processing an image, and issue an error message if the parameters are not internally consistent. Suppose you tried to use the File/New/Grayscale command to create a 1-by-1-by-128 grayscale with 256 different graylevels, you would get the error message shown below. To continue, dismiss the message by clicking on the OK button or by pressing the Return or Enter key, and re-issue the command with the parameters modified appropriately.



Several I/O-related MacSADIE error messages contain a numeric Macintosh System error code in addition to the message itself. For a listing of these error codes and their meanings, refer to Apple Computer Inc., *Inside Macintosh*, Vols. I–VI, Addison-Wesley, Reading, MA, 1985–1991.

The most common memory-related error message is shown below. It is displayed whenever MacSADIE cannot allocate a sufficient amount of memory. Keep in mind that memory is allocated not only for new images to be created — some commands (which are explicitly marked as memory-intensive) also need considerable amounts of memory temporarily in order to perform their function. If you encounter this message, you should first close any unneeded image windows, and re-issue the command you were trying to execute. If that does not solve the problem, you should save to disk and close all images but the one you are trying to process. If there still is not enough space available, it may be because of memory fragmentation. In that case, it is best to save the unprocessed image data to disk, quit and re-start MacSADIE. If you are running MacSADIE under MultiFinder, you can also increase MacSADIE's memory partition with the Get Info command from the Finder's File menu.



If you encounter any other memory-related problems, it is probably best to save your image data to disk, and then quit and re-start MacSADIE. If the problem persists (and can be reliably reproduced), you should file a report with the Digital Image Analysis Lab at the University of Arizona (see Appendix D).

Macros

MacSADIE is based on the "toolbox" concept, which means that each command performs a basic image processing function. However, for many applications it is desirable to be able to combine multiple functions into more complex image processing sequences. This can easily be done with so-called macro utilities, which let you define complex sequences of mouse movements, mouse clicks, and keyboard input, and "re-play" those sequences with a single keystroke. MacSADIE itself does not provide a macro capability, but is compatible with MacroMaker from Apple (which comes free of charge with the Macintosh system software), as well as with any of the commercially available macro packages for the Macintosh such as QuicKeys from CE Software or Tempo by Affinity Microsystems.

The Menu

The contents of the menu depend on the contents of your system file as well as on the current state of your desktop — which desk accessories you have installed, which application you are currently using, and whether you are running under Finder or MultiFinder.

The first item on the menu always provides you with important information on the current application — MacSADIE in the example on the right.

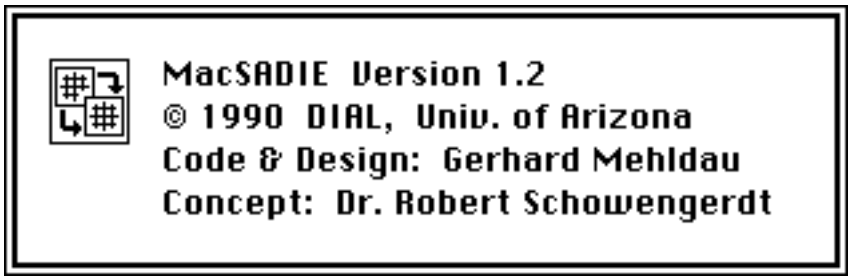
The second block of items on the menu lets you choose from a list of installed desk accessories.

If you are running MacSADIE under MultiFinder, the menu contains two additional blocks of choices — a list of all open applications, and an item on MultiFinder. The currently active application (MacSADIE in this example) is indicated by a check mark in front of its icon. Switching between applications can be accomplished by simply selecting another application — it causes the application currently in front to be put in the background, and the selected application is brought to the front.

Note that the contents and/or the organization of the menu may change with future versions of Apple's system software.

About MacSADIE

When you select this item, MacSADIE displays a dialog similar to the one shown below. It tells you which version of MacSADIE you are running, and gives you other important information on MacSADIE as well. To dismiss the dialog and continue, simply click anywhere on the screen, or press any key.

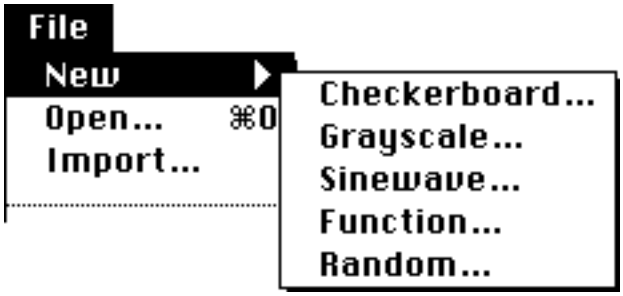


The File Menu

The File menu lets you create new images with the New command and read image data from disk in MacSADIE, PICT or TIFF format with the Open command, and in generic format with the Import command. You can dispose of an image using the Close command and write image data to disk with the Save and Save As commands. The Page Setup and Print commands let you format and print images or the contents of the session log window. To terminate a session and to exit from MacSADIE, select the Quit command.

The New, Open, Import and Quit commands are always available (enabled), whereas the Close, Save, and Save As commands are enabled only when the frontmost window contains a regular image (as opposed to the session log or clipboard windows). The Page Setup and Print commands are enabled only when either a window containing a regular image or the session log window is frontmost.

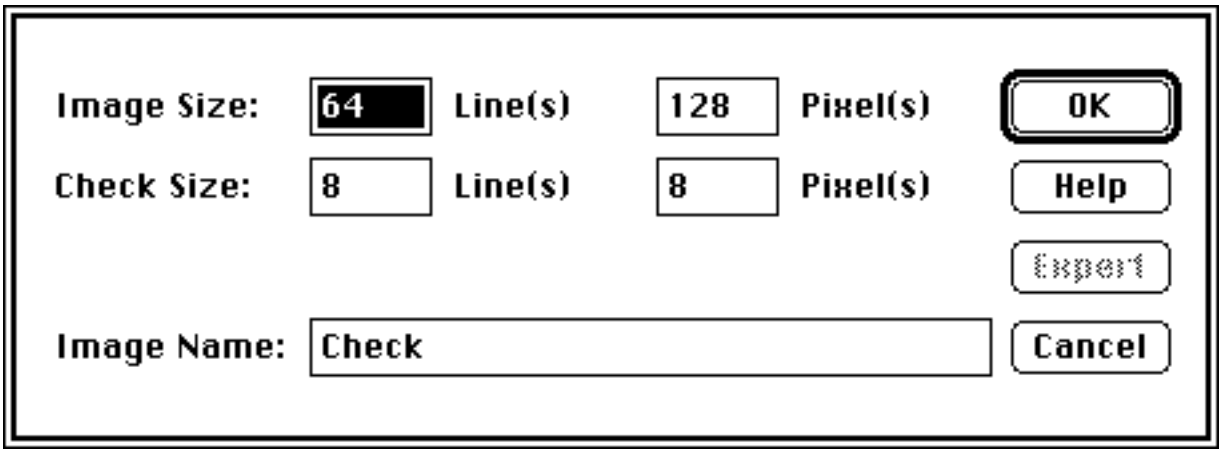
New



The New command provides, through a submenu, commands for creating synthetic images. Choose one of the five commands from the submenu, according to the type of image you want to create. All choices on the New submenu are permanently enabled.

Checkerboard

The Checkerboard command generates a binary checkerboard pattern, starting with a black rectangle (a graylevel of 0) in the upper left corner, and alternating between white and black (graylevels of 255 and 0, respectively) thereafter. Parameters are the number of lines and the number of pixels per line of the output image, and the vertical and horizontal extent of an individual rectangle.



The graylevel value of a pixel in the output image O is computed as

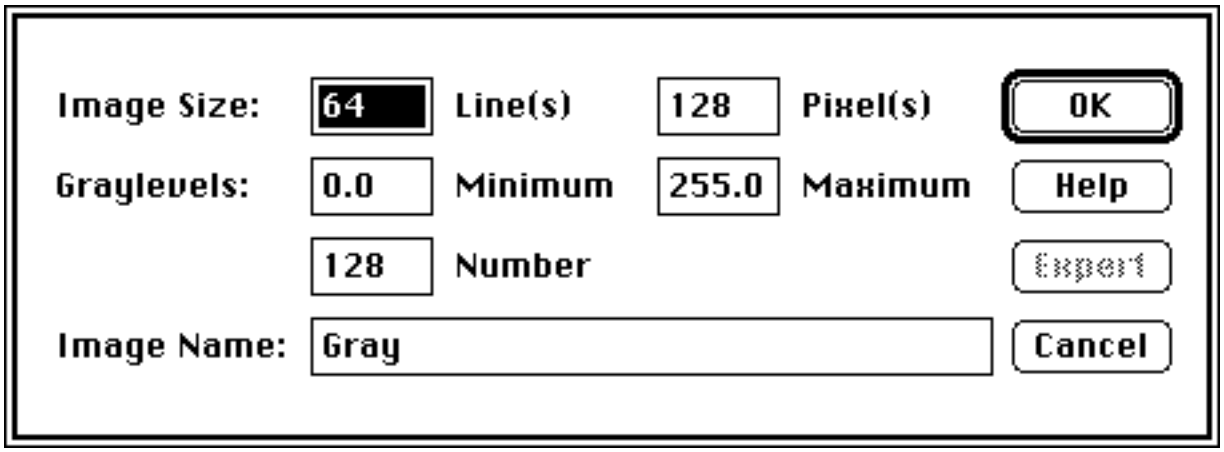
$$O_{jk} = \begin{cases} 0 & \text{if } \mathrm{mod}(\lfloor (j-1)/v \rfloor + \lfloor (k-1)/h \rfloor, 2) = 0 \\ 255 & \text{otherwise} \end{cases}$$

where mod() represents the modulo function, and v and h are the vertical and horizontal extents, respectively, of an individual rectangle, as specified in the dialog.

You can create a checkerboard pattern starting with a white rectangle in the upper left corner by doing a Contrast/Linear Stretch (with the appropriate parameters) on the original checkerboard. If you set the vertical check size equal to the number of lines in the image, the Checkerboard command creates a vertical bar pattern. Similarly, setting the horizontal check size equal to the number of pixels per line results in a horizontal bar pattern. A useful mask function may be created by doing a Contrast/Linear Stretch to minimum and maximum graylevels of zero and one, respectively.

Grayscale

The Grayscale command creates a horizontal grayscale pattern. Parameters are the size of the image to be generated, the minimum and the maximum graylevel values, and the number of distinct graylevels. The generated grayscale has the minimum graylevel on the left, and the maximum graylevel on the right side.



The graylevel value of a pixel in the output image O is computed as

$$O_{jk} = \frac{(k-1)}{n} \cdot \frac{g \cdot (O_{max} - O_{min})}{(g-1)} + O_{min}$$

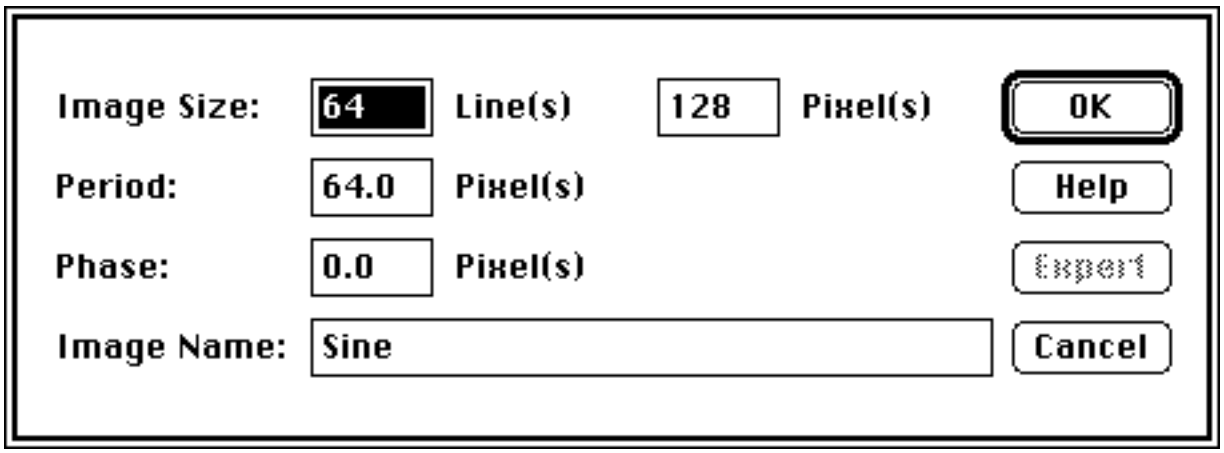
where n is the number of pixels/line, O_{min} and O_{max} are the minimum and maximum graylevels, and g is the number of graylevels in the output image, as specified in the dialog.

You can use the Geometry/Rotate command to reverse the order of graylevels in the output image, or to rotate the grayscale from horizontal to vertical. To append a grayscale to an existing image, you can use the Geometry/Horizontal Mosaic, the Geometry/Vertical Mosaic, or the Geometry/General Mosaic command. When doing this, you should keep in mind the automatic display stretch used by MacSADIE. In order for the combined image to utilize the full dynamic

range of the display, the grayscale being appended must have the same graylevel minimum and maximum as the original image.

Sinewave

The Sinewave command creates a horizontal sinewave pattern. Parameters are the size of the output image, and the period and phase of the sinewave.



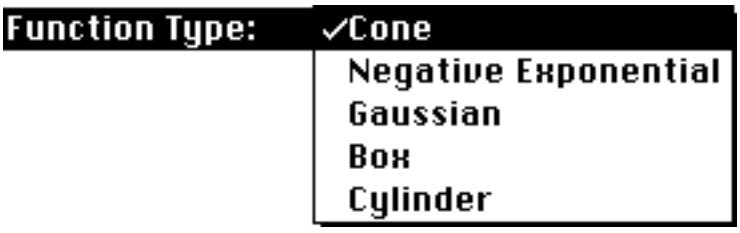
The graylevel value of a pixel in the output image O is computed as

$$O_{jk} = 255 \cdot \left(\sin\left(2\pi \cdot \frac{(k-1)}{(p-1)} + \varphi \right) + 1 \right) / 2$$

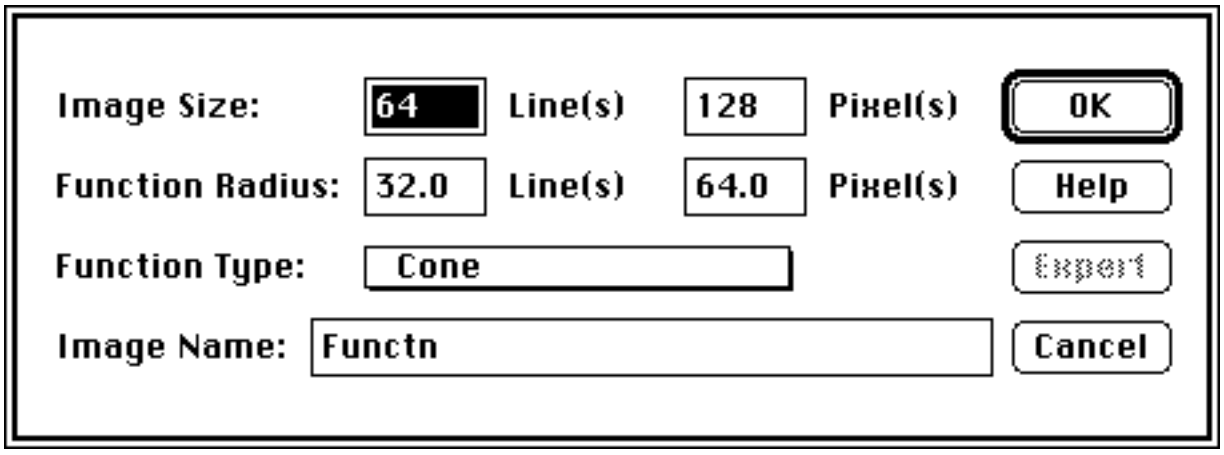
where p is the period and φ is the phase of the sinewave, as specified in the dialog, and sin() is the trigonometric sine function.

The minimum graylevel in the generated image is 0, the maximum is 255. You can change the amplitude and bias level of the sinewave with the Contrast/Linear Stretch command.

Function



The Function command lets you generate a variety of two-dimensional test functions. Parameters are the size of the generated image, the function's radii, and the function type, selected from a pop-up menu on the dialog. All choices on the pop-up menu are permanently enabled.



The graylevel value of a pixel in the output image *O* is computed for the various types of test functions as

Cone: $$O_{jk} = \begin{cases} 255 \cdot (1-r) & \text{if } r \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Negative Exponential: $$O_{jk} = 255 \cdot e^{-r}$$

Gaussian: $$O_{jk} = 255 \cdot e^{-r^2}$$

Box: $$O_{jk} = \begin{cases} 255 & \text{if } (j-j_c) \le r_j \wedge (k-k_c) \le r_k \\ 0 & \text{otherwise} \end{cases}$$

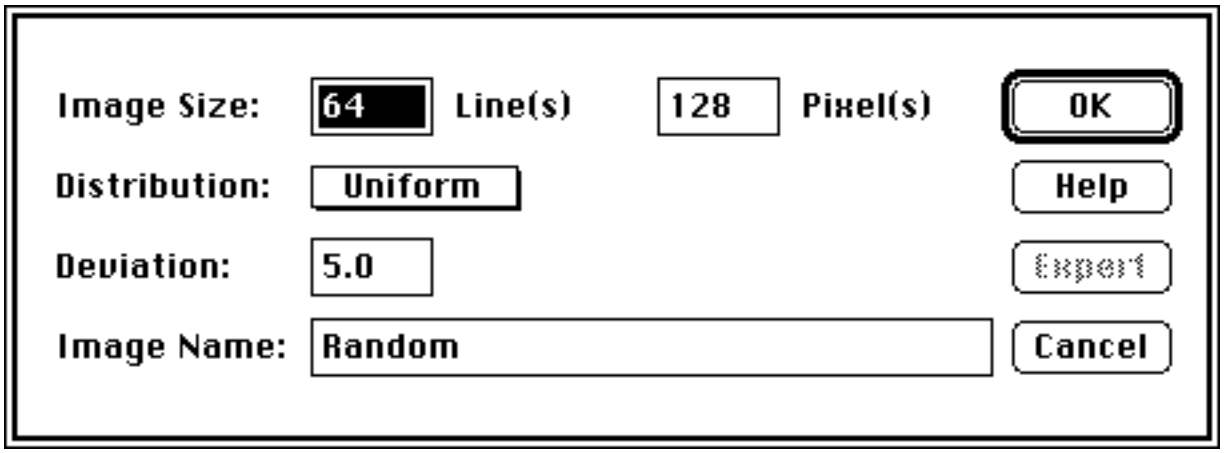
Cylinder: $$O_{jk} = \begin{cases} 255 & \text{if } r \le 1 \\ 0 & \text{otherwise} \end{cases}$$

where $r = \sqrt{((j-j_c)/r_j)^2 + ((k-k_c)/r_k)^2}$, $j_c = \lfloor m/2 \rfloor + 1$ and $k_c = \lfloor n/2 \rfloor + 1$ are the coordinates of the center of the test function, r_j and r_k are the radii of the test function, as specified in the dialog, and e is the base of the natural logarithm (i.e., $e = \ln(1) = 2.718282\ldots$).

The center of the test function in an image of size 1-by-*m*-by-*n* is located at pixel $(\lfloor m/2 \rfloor + 1, \lfloor n/2 \rfloor + 1)$, so that the image can be used as a kernel in a Filter/Convolution or Filter/FFT Convolution command.

Random

The Random command creates an image consisting of a pseudo-random pattern. Parameters are the size of the image to be generated, the probability distribution of the image's graylevel values, and the graylevel deviation, which corresponds to either the range (in the case of a uniform distribution) or the standard deviation (in the case of a gaussian distribution). The graylevel distribution is chosen from a pop-up menu, with all choices permanently enabled.



The graylevel value of a pixel in the output image O is computed as

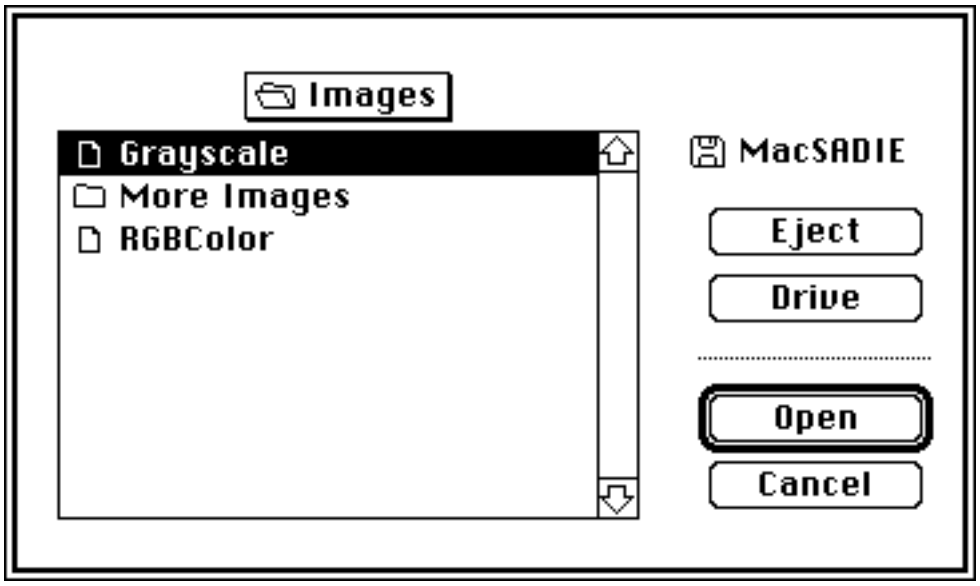
$$O_{ijk} = \begin{cases} d \cdot \left(\dfrac{R}{R_{max}} - 0.5 \right) & \text{(if Uniform Distribution is specified)} \\ d \cdot \left(\dfrac{\sum_{i=1}^{12} R_i}{R_{max}} - 6.0 \right) & \text{(if Gaussian Distribution is specified)} \end{cases}$$

where d is the deviation specified in the dialog, and R and R_i, $1 \le i \le 12$, are (uniformly distributed) random variables in the range $[0, R_{max}]$.

The Random command is useful for simulating image noise. You can use the Misc/Combine command to add the generated noise to an existing image.

Open

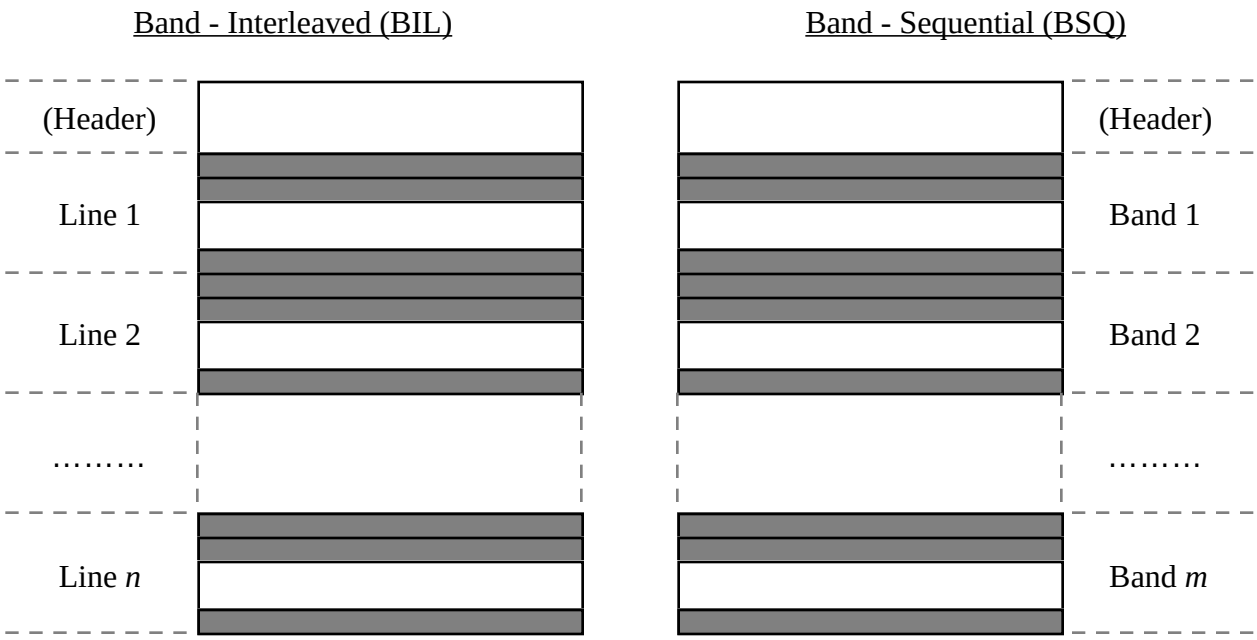
The Open command reads an image from disk and displays it on the desktop. MacSADIE can read images in its own format, as well as PICT and TIFF images. The dialog presented by the Open command is a standardized dialog used by many applications. On top of its main window it displays the name of the current folder. The main window itself shows the contents of this folder, that is, all MacSADIE, PICT and TIFF images, as well as any sub-folders located in the current folder. You can traverse the hierarchy of folders on all available disks just as you would with any other Macintosh application. To select and open an image, either double-click on the file name, or select the file and click on the Open button.



Holding down the Shift key when selecting the Open command has no effect, i.e., it does not cause the dialog to be bypassed. Note that opening a PICT image is a memory-intensive operation.

Import

The Import command lets you read images created by applications other than MacSADIE, or images transferred to the Macintosh from other computers (“generic” images), if they conform to one of the following two formats (note that the formats are equivalent for a single-band image).



The diagram assumes an *m*-by-*n*-by-*o* image, where

is one line (*o* pixels) of image data from Band 1,

is one line (*o* pixels) of image data from Band 2,

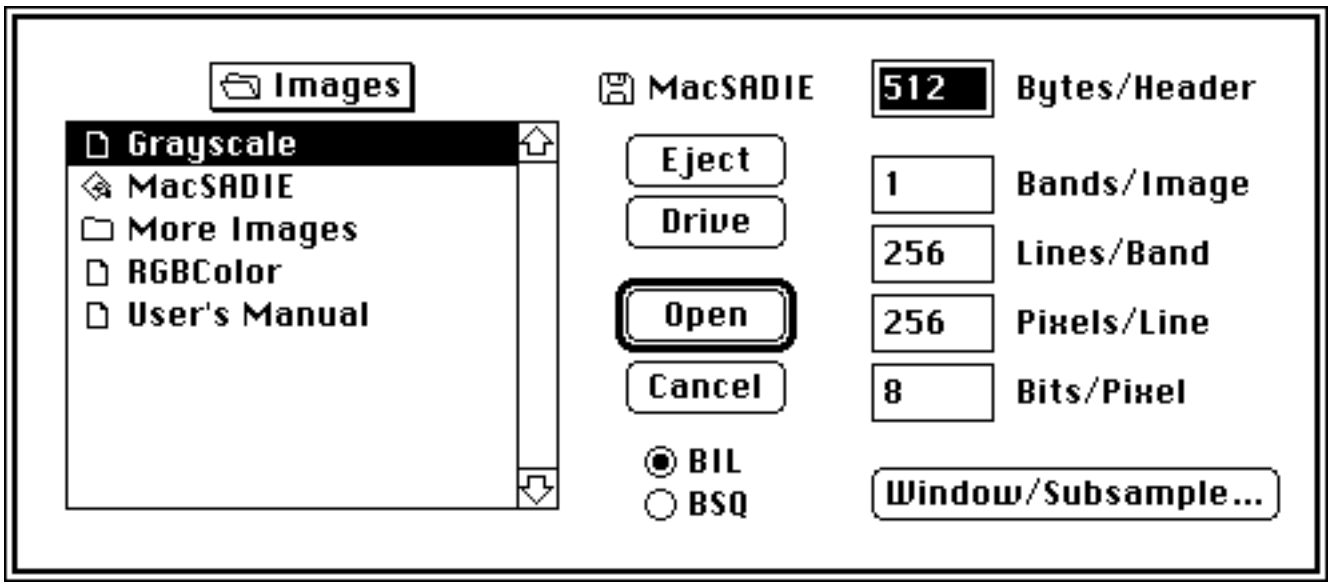
and

is one line (*o* pixels) of image data from Band *m*.

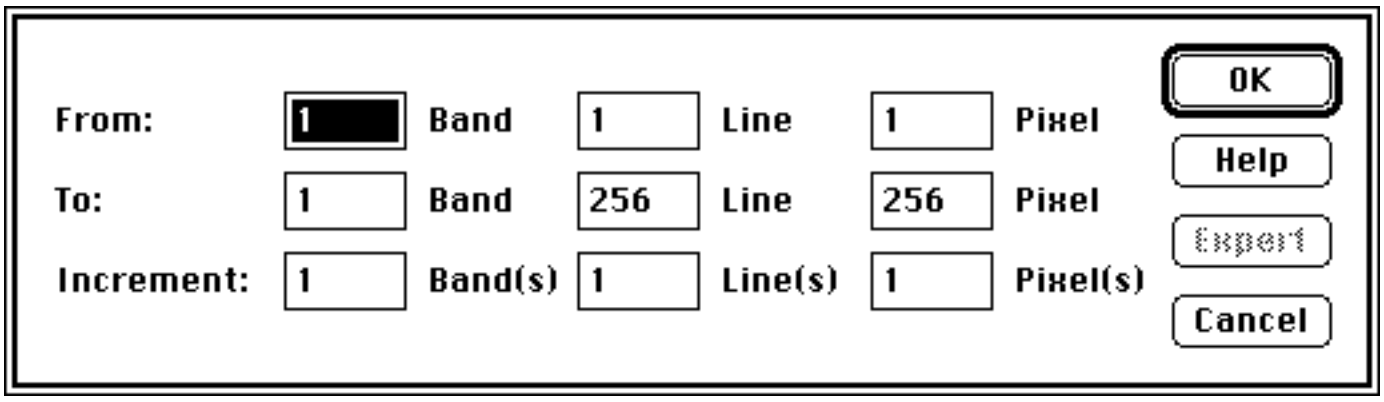
In addition to the controls from the Open command's dialog, the dialog for the Import command requires you to specify the parameters of the two formats above. The format itself is chosen with the BIL and BSQ radio buttons, respectively. The size of the header (which may be between 0 and 32K Bytes long), the size of the image, and the number of bits/pixel (between 1 and 31 for integral values, or 32 for floating point values in Macintosh format) must be specified explicitly. The default header size is set for MacSADIE files (see Appendix C for the MacSADIE file format).

Note that the main window not only lists sub-folders and MacSADIE images, but all types of files located in the current folder. Needless to say, the results may be unexpected if the imported file does not contain an image of the specified format. In particular, if the specified format does not exactly match the structure of the disk file, a variety of I/O-related error messages may result.

As with the Open command, the Shift key does not have any effect with the Import command.



To read in only part of an image, first specify the full image size as described above and then click on the Window/Subsample… button. You will then find the secondary dialog shown below, where you can specify the area to be windowed and/or subsampled. This area defaults to the entire image as specified in the primary dialog. This option is useful if you want to view or process only part of the image, or if the entire image is too large to fit into available memory.



Close

The Close command removes the frontmost image window from the desktop and returns all memory associated with it to the system. **MacSADIE does not ask whether you want to save a previously unsaved image before it closes that image!** This "feature," which runs counter to most Macintosh applications, was implemented for two reasons. Most images created with MacSADIE are of a temporary nature, i.e., they represent intermediate results in a chain of processing steps, and are normally discarded after some "final" image has been produced. Since digital images take up large amounts of disk space, MacSADIE actively discourages users from storing these intermediate images in this manner. Furthermore, since a particular image can not

be modified (an image processed with some command always results in a new image), it can easily be reproduced by re-applying the processing sequence to the original image.

Clicking on the close box of a regular image window (in the upper left corner) is equivalent to choosing the Close command. If you hold down the Command key ( ) when clicking on the close box, all MacSADIE image windows on the desktop are closed, and the session log and clipboard windows, if visible, are hidden. This feature is useful if you want to clean up your desktop before you switch to another application under MultiFinder, but don't want to quit MacSADIE entirely.

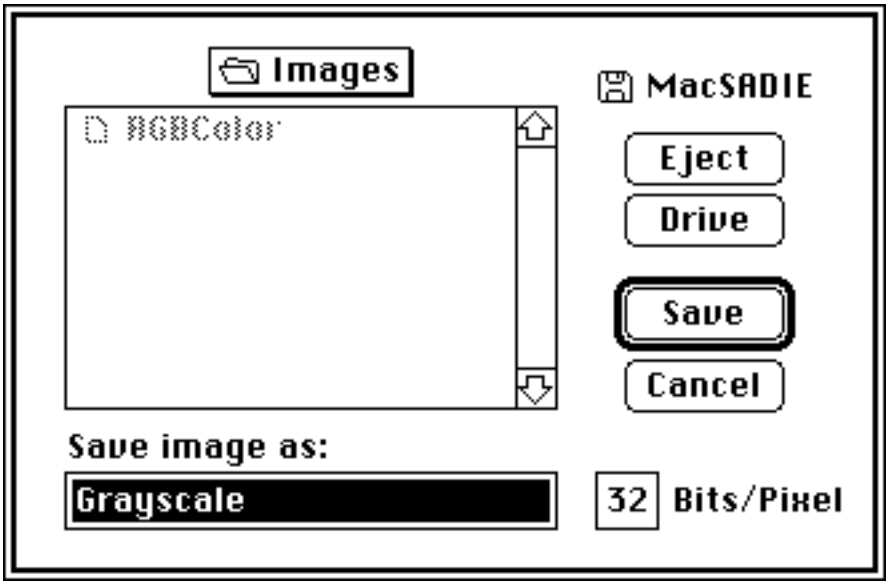
Save

The Save command lets you write an image to disk in MacSADIE's file format. When the Save command is invoked for the first time on a particular image, it prompts you with the Save As command's dialog (see below) to let you specify the file's name and location on disk, as well as the number of bits to be used to store the graylevel of an individual pixel.

Since images on the desktop can not be modified in MacSADIE, re-saving an already saved image is redundant. Therefore, MacSADIE does nothing if the Save command is invoked more than once on the same image. Use the Save As command if you want to create a copy of an already saved image.

Save As

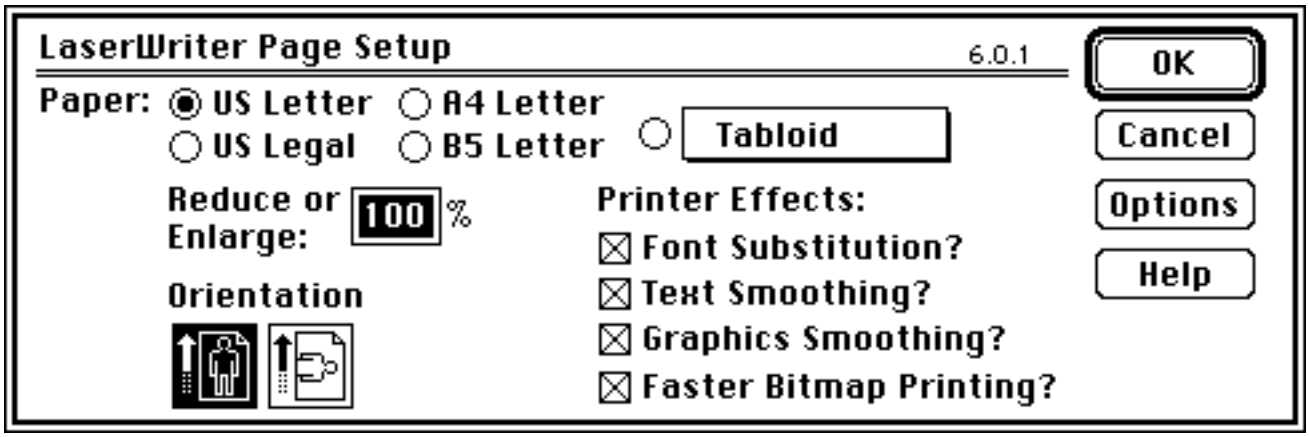
The Save As command lets you write an image to disk in MacSADIE's file format. The dialog again is a Macintosh standard, and is similar to the Open dialog — you can navigate through the hierarchy of folders on all available disks in the same fashion. The file name defaults to the name of the image to be saved. If you try to save an image with the same name as an already existing file, MacSADIE prompts you for confirmation of your choice before overwriting the old file. To prevent any loss of accuracy, graylevel values are written to disk as floating point quantities by default.



With the number of bits/pixel set to 32, graylevel values are written to disk in floating-point format. To reduce the disk space required to store an image, you can set the number of bits/pixel to any number n, $1 \le n \le 31$, if the image graylevel values are in the range $[0,2^n)$. Each pixel is then converted to an integer in the range $[0,2^n-1]$ (by having its fractional part truncated), and written to disk in n-bit integer format.

Page Setup

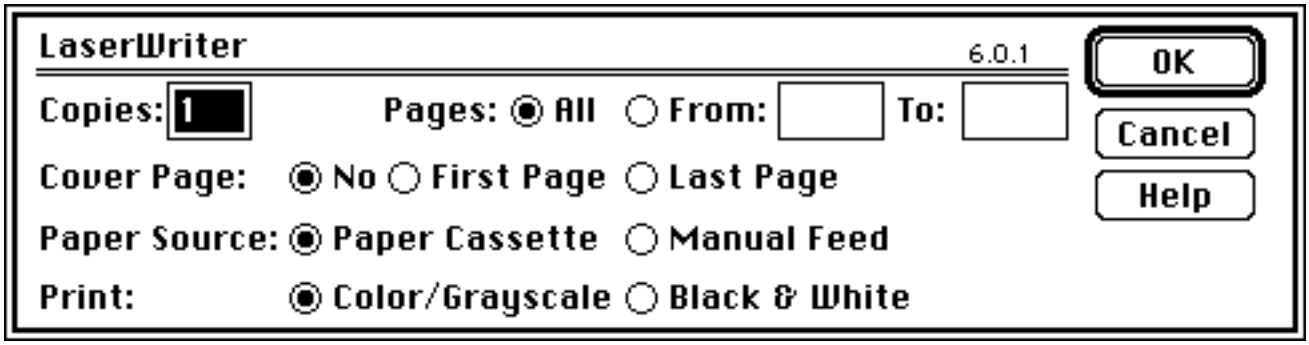
The Page Setup command lets you control the format of your printout. The Page Setup command's dialog is not specific to MacSADIE; rather, the formatting options available depend on the particular type of printer you have chosen. The dialog below shows the case of a LaserWriter.



All formatting parameters as specified with the Page Setup command are retained with the document (image or session log) during a single session — it therefore suffices to specify the format of the printout once per document. Holding down the Shift key when selecting the Page Setup command has no effect, i.e., it does not cause the dialog to be skipped.

Print

The Print command lets you create a hardcopy of an image or of the session log. As with the Page Setup command, the Print command's dialog depends on your particular system configuration. Shown below is the case of a LaserWriter.



Images are printed centered and without header on a single page, truncated if necessary. You can use the Page Setup command to scale the image for printing, or to change the orientation of the image with respect to the page. The print quality (color or grayscale, dithered or halftoned) depends on the capabilities of your printer and its software. Note that printing a halftoned image may be a time-consuming operation, and the use of a print spooler such as Print Monitor from Apple (which comes free of charge with the Macintosh system software), or any of the commercially available spoolers for the Macintosh (such as SuperSpool or SuperLaserSpool from SuperMac Technology, TurboSpool from Peripheral Land Inc., LaserServe from Infosphere, or Grappler Spooler from Orange Micro), is highly recommended.

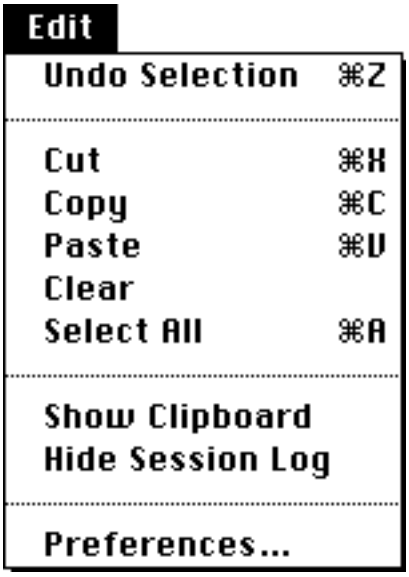
The session log is printed on a single or on multiple pages, dependent on its length. You can obtain and print additional information if you use the Quit command (see below) to print out the session log.

Quit

The Quit command terminates a MacSADIE session. It closes all image windows on the desktop and hides the session log and clipboard windows, if they are visible.

Holding down the Shift key when choosing the Quit command has an effect that is different from its effect on other commands. First, the session log window is made visible (if it is not already) and statistical data on the current session (SADIE functions called, number of calls, absolute and relative time spent in each function) is added. After that, MacSADIE brings up the dialog for the Print command (see above) to let you print the modified session log. If you want to use a format other than the default for the printout, you should specify that format (using the Page Setup command) before choosing the Quit command. Also note that clicking on the Cancel button in the Print dialog only cancels printing; it does not cancel the Quit command.

The Edit Menu



The Edit menu lets you exchange images and data between different applications via the clipboard, and provides commands for changing the visibility of the clipboard and session log windows. The Preferences command has not yet been implemented.

The Undo/Redo command is enabled whenever the last edit command is undoable. The Cut and Copy commands are enabled only when the frontmost window contains a regular image (i.e., not for the clipboard or session log windows), and when the selection rectangle of the image in the frontmost window is non-empty. The Paste command is enabled whenever the clipboard is not empty. The Clear and Select All commands are enabled when the frontmost window contains a regular image, independently of the selection rectangle. The Hide/Show Clipboard, Show/Hide Session Log and Preferences commands are always enabled.

Undo / Redo

The Undo/Redo command is a single, toggled command, which lets you undo the last edit operation. The command normally reads Undo *op*, where *op* is the last edit operation (Cut, Copy, Paste, Clear, Selection) you performed. When you select Undo, this operation is undone, and the command changes to read Redo *op*. Redo *op* re-applies operation *op*, i.e., it undoes the effects of the Undo command. The window on which the Undo/Redo command operates is brought to the front, if it is not already there. If the last edit operation cannot be undone (for example, if you make a selection within some window, and then close that window, or if you copy a selection to the clipboard, and then replace the contents of the clipboard from within some other application), the Undo command reads Can’t Undo and is disabled.

Cut

The Cut command copies the contents of the selection rectangle of the frontmost window to the clipboard, and then sets that selection rectangle to the empty rectangle. Invoking the Cut command is equivalent to choosing the Copy command, followed by the Clear command. Note that the Cut command performs a memory-intensive operation.

Copy

The Copy command copies the contents of the selection rectangle of the frontmost window to the clipboard. Note that the Copy command performs a memory-intensive operation.

Paste

The Paste command creates a new, untitled window, and copies the contents of the clipboard into it. Note that, due to MacSADIE’s display stretch, a pasted image may look different from the original; however, graylevel values are preserved if the clipboard contents stem from a MacSADIE image.

The Paste command may be used to create an image of text from the clipboard (e.g., by copying from the Note Pad desk accessory or from a word processor document). This text image can then be used to annotate a "regular" image by combing the two, using one of several commands available from the Geometry and Misc menus.

Clear

The Clear command sets the selection rectangle of the frontmost window to the empty rectangle.

Select All

The Select All command sets the selection rectangle of the frontmost window to cover the entire image in that window.

Show Clipboard / Hide Clipboard

The Show/Hide Clipboard command is a single, toggled command. If the clipboard window is visible, the command reads Hide Clipboard, and executing it causes the window to be hidden. If the window is hidden, the command reads Show Clipboard, and executing it makes the window visible again. Clicking on the close box of the clipboard window is equivalent to choosing the Hide Clipboard command from the Edit menu.

Note that hiding a window is different from closing it. All information in the window is retained while the window is invisible. The clipboard window is continually updated as the contents of the clipboard change, independently of its visibility.

Hide Session Log / Show Session Log

The Hide/Show Session Log command is a single, toggled command. If the session log window is visible, the command reads Hide Session Log, and executing it causes the window to be hidden. If the window is hidden, the command reads Show Session Log, and executing it makes the window visible again. Clicking on the close box of the session log window is equivalent to choosing the Hide Session Log command from the Edit menu.

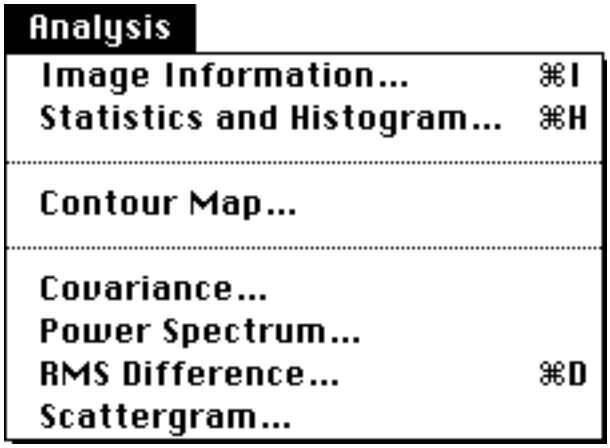
Note that hiding a window is different from closing it. All information in the window is retained while the window is invisible. In the case of the session log, new information is added independently of the window's visibility.

Preferences

The Preferences command has not yet been implemented. We welcome any and all suggestions from our users as to which features of MacSADIE you would like to be able to customize. Please send your ideas to:

MacSADIE c/o Digital Image Analysis Lab
Dept. of Electrical and Computer Engineering
University of Arizona
TUCSON, AZ 85721, U.S.A.
macsadie@dialsun.ece.arizona.edu

The Analysis Menu

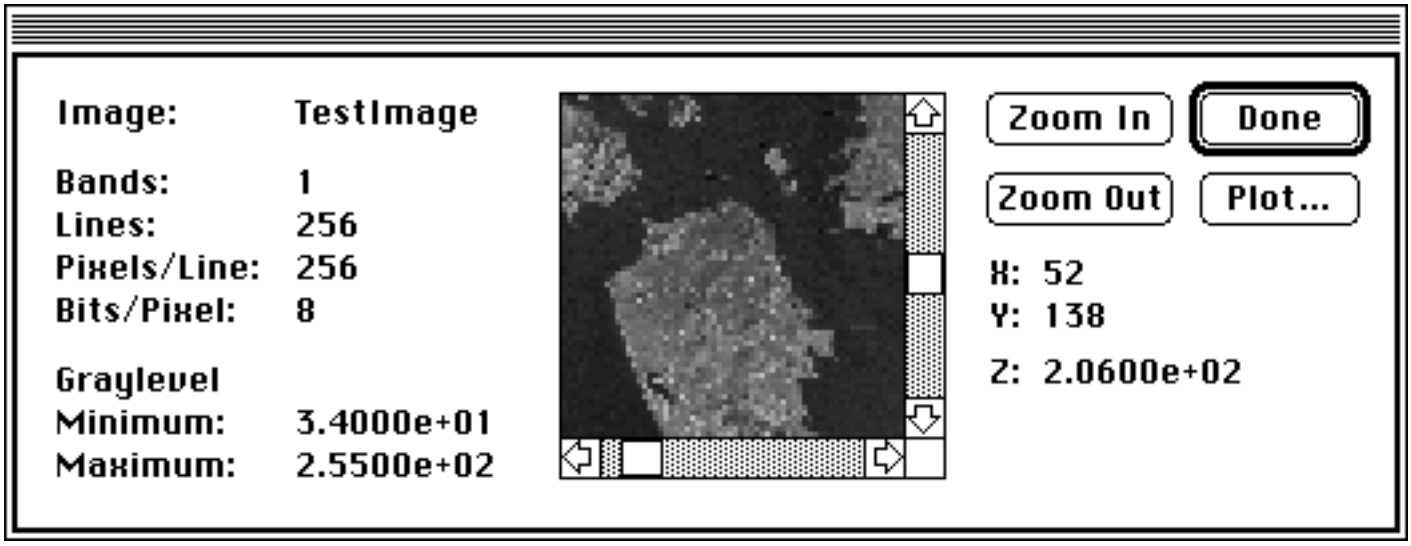


The Analysis menu provides you with a variety of commands for statistical and other types of image analyses. These commands are useful for deciding which type of image processing function to apply to a particular image, or what parameters should be used with a given processing function.

The Image Information and Statistics and Histogram commands are enabled for all types of images, whereas the Contour Map and Power Spectrum commands are enabled only for single-band images. The Covariance command only operates on multi-band images, and is therefore enabled only when the frontmost window contains a multi-band image. The RMS Difference command compares two images, and is enabled whenever there are two or more image windows on the desktop. The Scattergram command operates on two bands from either the same or two different images, and is therefore enabled only when at least two different bands are available.

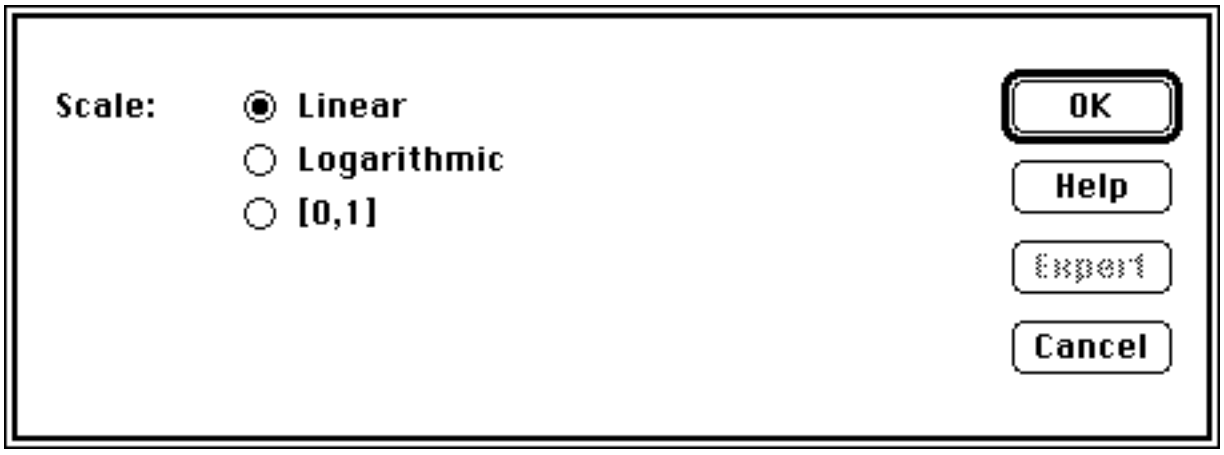
Image Information

The Image Information command uses a type of dialog that is different from most MacSADIE commands. This dialog does not prompt you for parameters, nor does it initiate any processing, but instead lets you interactively "explore" the image. Besides listing general image attributes, it lets you examine the graylevel values of individual pixels and arbitrary cross-sections of the image. You can move the dialog around the screen just like an ordinary window, and, under MultiFinder, you can switch between applications while the dialog is displayed. To dismiss the dialog, click on the Done button, or press the Return or Enter key.



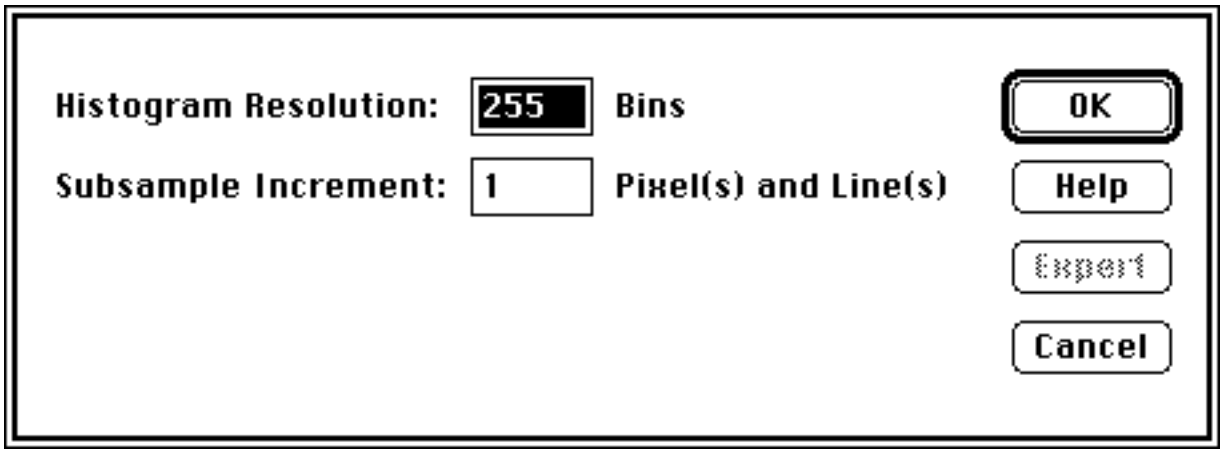
On the left side, the dialog shows the name of the image, its size, and its graylevel minimum and maximum. In the center, the image itself is displayed in a 128-by-128 window. You can scroll the image through the window by using the scroll bars to the right and underneath, and zoom with the Zoom In and Zoom Out buttons to the right. The minimum (and default) zoom is 1:1 (i.e., one image pixel corresponds to one pixel in the window), and the maximum zoom is 1:128 (i.e., one image pixel fills the entire window), with steps of 1:2, 1:4, 1:8, 1:16, 1:32, and 1:64 in between. The Zoom In (Zoom Out) button is disabled once the maximum (minimum) zoom has been reached. You can examine individual pixels by moving the cursor over the image. When the cursor is on top of the image, its shape changes to a crosshair, and its location (*X* and *Y* in image coordinates) is displayed together with the graylevel value(s) at that location (*R*, *G*, and *B* for three-band (color) images, and *Z* for single-band and multi-band images displayed in grayscale format).

To define a cross-section, move the cursor to one endpoint of the line, push (and hold down) the mouse button, drag the cursor to the other endpoint and release the mouse button. The image will scroll automatically if you move the cursor outside the window. Once defined, a cross-section persists across scrolls and zooms until you either clear it by clicking on the window, or until you define another cross-section. To get a profile of the graylevel values of the currently defined cross-section, click on the Plot… button (which is enabled only when a cross-section is defined). You can then chose the scale for the profile's graylevel axis from the secondary dialog shown below. The Logarithmic option is enabled only if all pixels along the profile have non-negative graylevel values. You can skip over the secondary dialog by holding down the Shift key when clicking on the OK button.



Statistics and Histogram

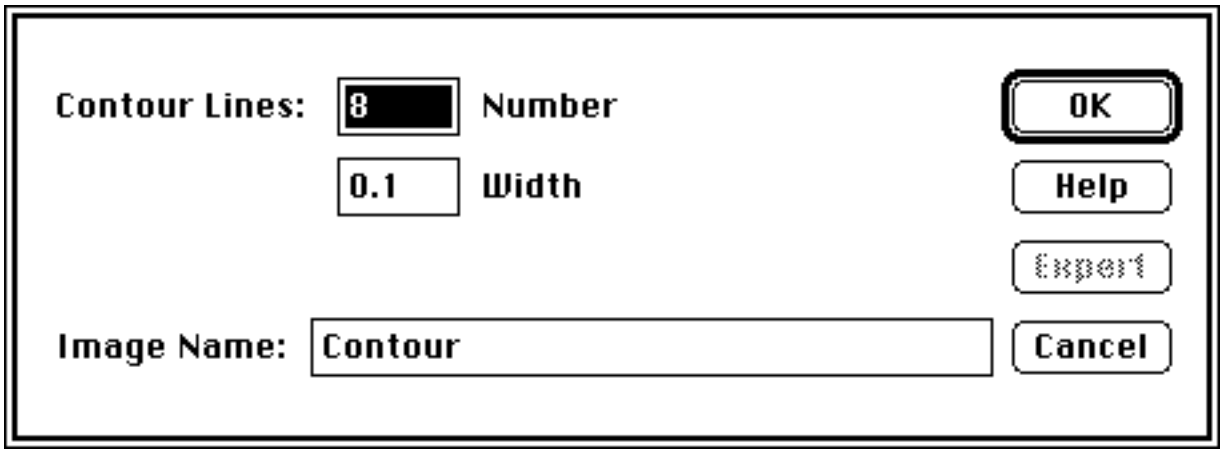
The Statistics and Histogram command computes, for each band of the image, the mean graylevel and its standard deviation, the minimum and maximum graylevels, the graylevel range, and the number of zero and non-zero pixels in that band. In addition, the command optionally computes and draws in the sesson log the histogram of graylevels for each band. Parameters are the number of bins to be used in the histogram and the pixel and line increment for sampling in each band. The default number of histogram bins is computed from the input image's graylevel minimum and maximum (I_{min} and I_{max}) as $\lfloor I_{max} \rfloor - \lfloor I_{min} \rfloor$. If the number of bins is set to zero, no histogram is computed.



The subsample increment parameter allows you a tradeoff between accuracy and speed. A smaller value uses more pixels to compute the statistics and is therefore relatively slow; a larger value uses fewer pixels and is faster. Note, however, that the resulting values only approximate the correct statistics if the subsample increment is greater than one — in particular, the exact image minimum and maximum graylevels may be missed. Also note, that because of MacSADIE's representation of graylevel values as floating point quantities, the default value for the number of histogram bins may be a poor choice, and may need to be overwritten. The size and shape of an image's histogram(s) are important decision criteria for many image processing steps — most notably the type and parameters of a contrast stretch. For more details see the sections on individual contrast stretches in the next chapter.

Contour Map

The Contour Map command creates, from a single-band input image, a binary output image consisting of white contour lines (lines of equal graylevel values) on a black background. Parameters are the number and width of the contour lines. The width of a contour line is specified as a fraction (a value in the range $[0.0,1.0]$) of the spatial width of the adjacent classes.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

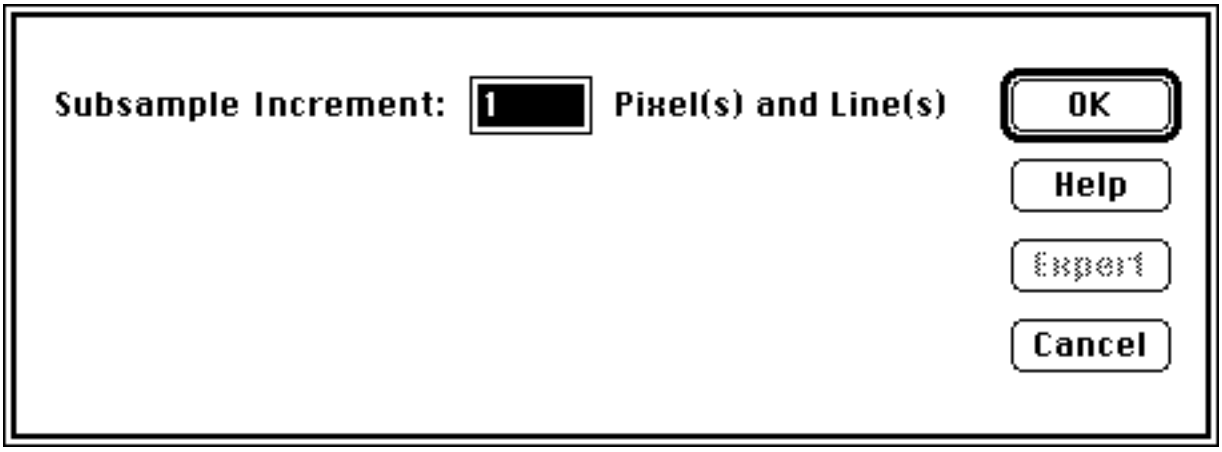
$$O_{jk} = \begin{cases} 0 & \text{if } \left| n \cdot \frac{I_{ijk} - I_{min}}{I_{max} - I_{min}} - \text{rnd}\left(n \cdot \frac{I_{ijk} - I_{min}}{I_{max} - I_{min}} \right) \right| \leq \frac{1 - w}{2} \\ 255 & \text{otherwise} \end{cases}$$

where n is the number of contour lines, and w is their width, as specified in the dialog, and the function rnd() rounds its real-valued argument to the nearest integer.

The output image is created by dividing the graylevel range of the input image into n+1 subranges (classes) of equal width, where n is the number of contour lines. Each pixel from the input image is assigned to one of those classes. The spatial boundaries between the classes form the contour lines.

Covariance

The Covariance command computes the covariance and correlation matrices among the bands of a multi-band image. The subsample increment parameter lets you define the pixel and line increment for graylevel sampling.



The covariance and correlation matrices for an m-by-n-by-o image I are defined, respectively, as

$$M_{COV} = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1m} \\ \vdots & \ddots & \vdots \\ \sigma_{m1} & \cdots & \sigma_{mm} \end{bmatrix}$$

$$M_{COR} = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1m} \\ \vdots & \ddots & \vdots \\ \rho_{m1} & \cdots & \rho_{mm} \end{bmatrix}$$

where σ_{ij} =

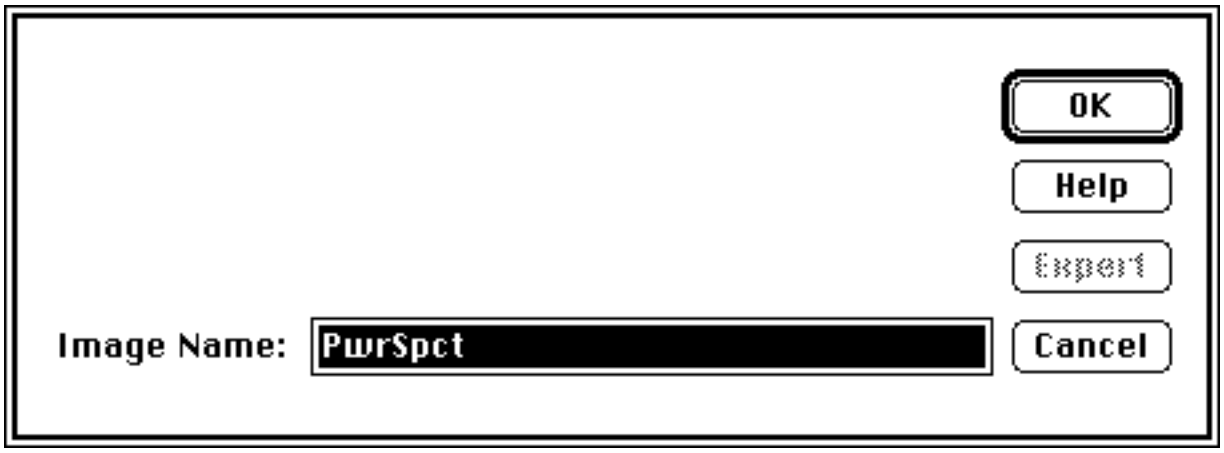
$$\begin{cases} \left(\sum_{k=1}^{n} \sum_{l=1}^{o} I_{ikl}^2 - \frac{\sum_{k=1}^{n} \sum_{l=1}^{o} I_{ikl} \cdot \sum_{k=1}^{n} \sum_{l=1}^{o} I_{jkl}}{n \cdot o} \right) / (n \cdot o - 1) & \text{if } i = j \\ \left(\sum_{k=1}^{n} \sum_{l=1}^{o} I_{ikl}^2 + \sum_{k=1}^{n} \sum_{l=1}^{o} I_{ikl} \cdot I_{jkl} - \frac{\sum_{k=1}^{n} \sum_{l=1}^{o} I_{ikl} \cdot \sum_{k=1}^{n} \sum_{l=1}^{o} I_{jkl}}{n \cdot o} \right) / (n \cdot o - 1) & \text{otherwise} \end{cases}$$

and $\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \cdot \sigma_{jj}}}$.

As with the Statistics and Histogram command, the subsample increment allows you a tradeoff between accuracy and speed. The diagonal terms of the covariance matrix represent the graylevel variances in the different bands, whereas the off-diagonal terms numerically quantify the amount of graylevel correlation between the bands. Note that both the covariance and correlation matrices are symmetric with respect to the main diagonal.

Power Spectrum

The Power Spectrum command computes the power spectrum (the squared modulus of the Fourier transform) of the input image, which must be single-band.



The graylevel value of a pixel in the output image *O* is computed as

$$O_{jk} = \mathrm{re}(\mathrm{FFT}(I)_{jk})^2 + \mathrm{im}(\mathrm{FFT}(I)_{jk})^2$$

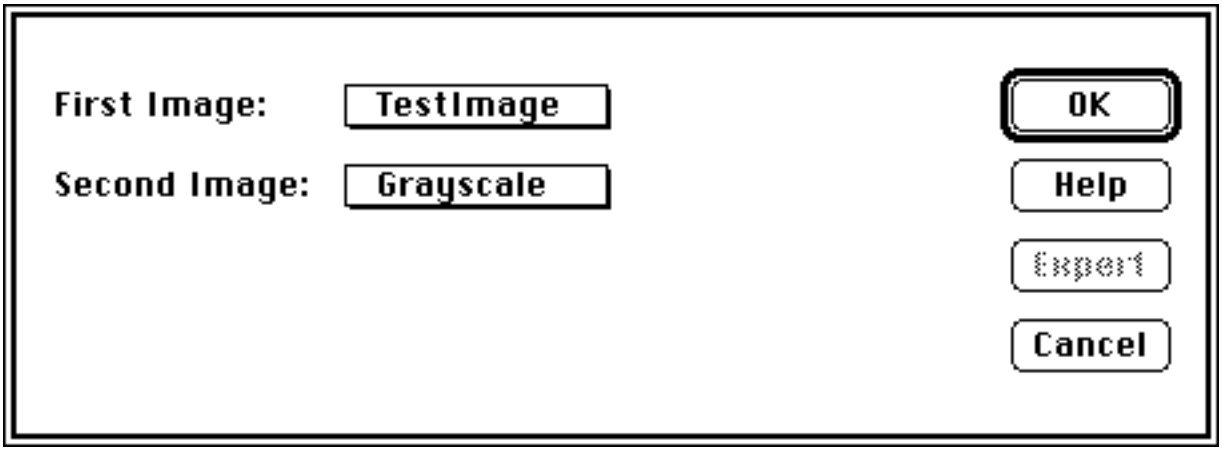
where *I* is the input image, FFT() is the discrete Fourier transform (see the description of the Filter/FFT command for details), and functions re() and im() deliver the real and imaginary parts of a complex number, respectively.

The typically large dynamic range in the image power spectrum usually benefits in visual display from contrast stretches such as Contrast/Logarithmic Stretch, or Contrast/Square Root

Stretch. The Power Spectrum command is useful for detecting periodic image or noise patterns. Note that computing the Power Spectrum of an image is a memory-intensive operation.

RMS Difference

The RMS Difference command computes the root-mean-squared (RMS) difference between two images of the same size. The dialog lets you choose the images to be compared via two pop-up menus, with all choices permanently enabled. The first image defaults to the image in the frontmost window, the second image to the one directly behind it. The two images must have the same number of bands, lines/band, and pixels/line.



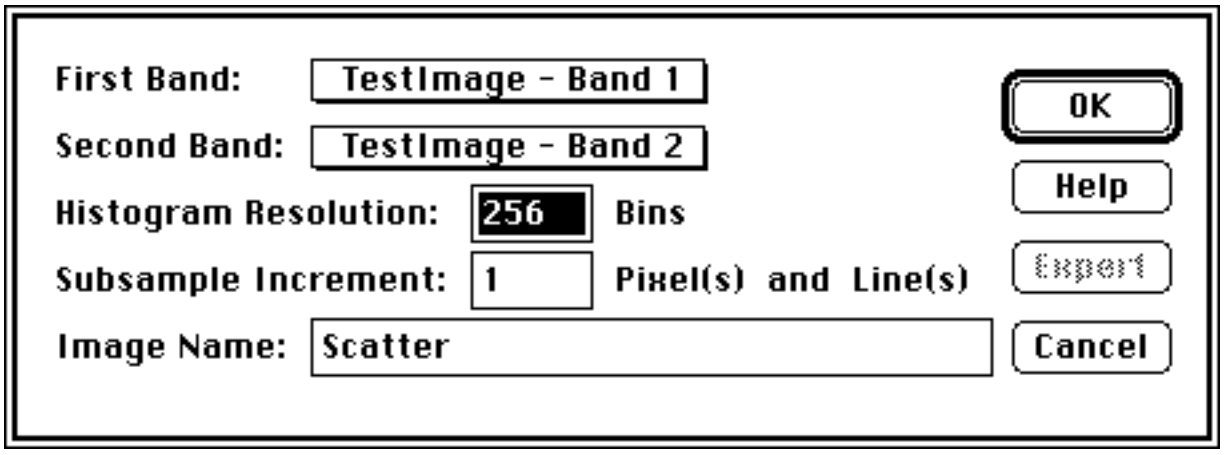
The RMS difference between two *m*-by-*n*-by-*o* images *I* and *J* is defined as

$$d_{RMS} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{o}(I_{ijk}-J_{ijk})^2}$$

The RMS difference represents an "average difference" between corresponding pixels in two images. It is equal to zero for two identical images, and increases as the two images become more and more dissimilar. The RMS Difference command is useful for comparing the similarity of two images, or the quality of a processed image relative to a reference image (e.g., in a simulation of sensor degradation and image restoration). It does not, however, correlate particularly well with visual image quality differences.

Scattergram

The Scattergram command creates an output image representing a two-dimensional histogram ("scattergram") between two bands. The bands may be from either the same or two different images, but must have the same number of lines and pixels/line. They are chosen from two pop-up menus which contain an entry for each band of every image on the desktop. All choices on the pop-up menus are always enabled. The first band defaults to the first band of the image in the frontmost window. If this is a multi-band image, the second band defaults to the image's second band; otherwise, it defaults to the first band of the image in the window behind the frontmost one. Additional parameters are the number of bins to be used along both histogram axes, and the pixel and line increment used for sampling in both images. The output image is of size 1-by-*n*-by-*n*, where *n* is the number of histogram bins.



The graylevel range of each input image is divided into n subranges of equal size, where n is the number of histogram bins. The graylevel value of a pixel in the output image, O_{jk}, is then computed as the number of pixel locations (m,n) in the input images I and J that have graylevel values in the jth and kth subrange, R_j and R_k, respectively:

$$O_{jk} = \left|\{(m,n): I_{mn} \in R_j \wedge J_{mn} \in R_k\}\right|$$

As with the Statistics and Histogram command, the subsample increment allows you a tradeoff between accuracy and speed. Note that the coordinate origin (1,1) of the scattergram image is at the upper left corner, consistent with all other MacSADIE images. The typically large dynamic range of image scattergrams usually benefits in visual display from contrast stretches such as Contrast/Logarithmic Stretch, or Contrast/Square Root Stretch. The Scattergram command is useful for visualizing the correlation between graylevels in different bands of a multi-band image prior to transforms such as the PCT (see Misc/Principal Components Transform) that de-correlate the different bands.

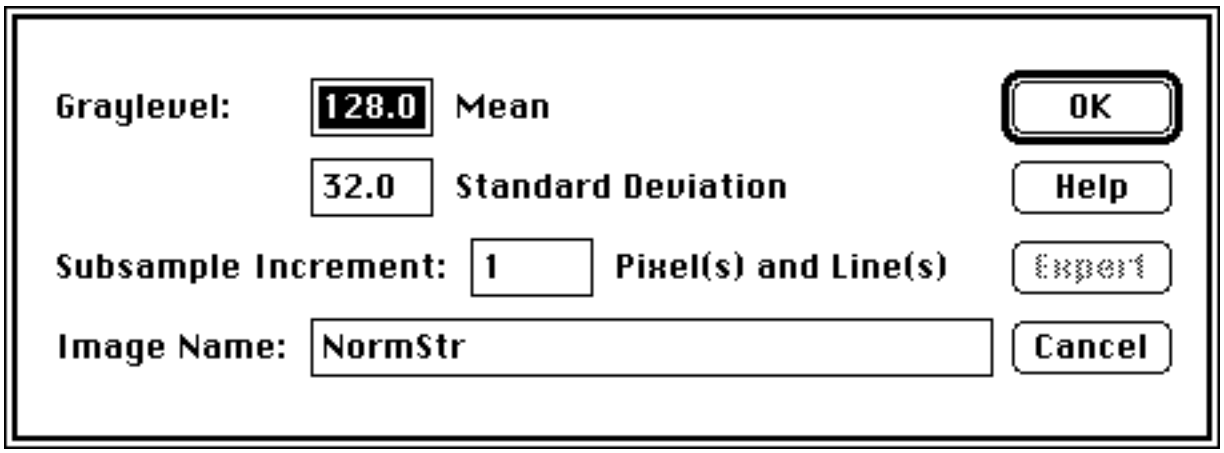
The Contrast Menu

The Contrast menu lets you choose from a large selection of commands for image contrast manipulation. Contrast stretches are point operations (i.e., the value of a pixel in the stretched image is a function only of the corresponding pixel in the original image) designed to enhance the image for visual interpretation. The image histogram often is a good indicator for deciding which kind of contrast stretch to apply.

With the exception of the Adaptive Stretch and Reference Stretch commands, all items on the Contrast menu are enabled for all types of images. The Adaptive Stretch command only operates on single-band images and is enabled only when the frontmost window contains such an image. The Reference Stretch command modifies the graylevels of one image dependent on those of a second image, and is therefore enabled only when there are at least two images available.

Normalize Stretch

The Normalize Stretch command performs a linear graylevel transformation on the input image. Parameters are the graylevel mean and variance in the output image, and the pixel and line increment used for sampling when the histogram of the input image is computed.



The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

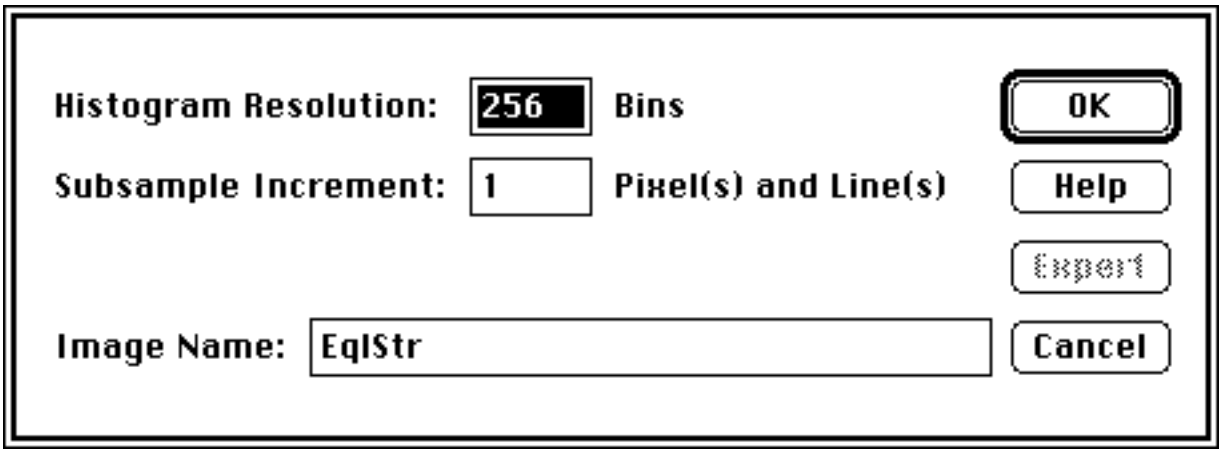
$$O_{ijk} = \frac{\sigma_O}{\sigma_I} \cdot (I_{ijk} - \mu_I) + \mu_O$$

where the input image has graylevel mean μ_I and standard deviation σ_I. The mean value of the output image, μ_O, and its standard deviation σ_O are the parameters specified in the dialog.

The subsample increment parameter allows you a tradeoff between accuracy and speed. A smaller value uses more pixels to compute the histogram and is therefore relatively slow; a larger value uses fewer pixels and is faster. Note that, due to the fact that MacSADIE stretches the graylevel range of each image individually before displaying it, an image and its linearly stretched version look identical on the screen. The Normalize Stretch command is useful for visually comparing images by adjusting them to have the same second order statistics. Because of the automatic display stretch used by MacSADIE, however, the comparison can only be done if both images are then processed with the Contrast/Saturate Graylevels command, with the same parameters for both images.

Histogram Equalize Stretch

The Histogram Equalize Stretch command transforms the image histogram to one for which the histogram density (number of pixels per graylevel) is approximately uniform. Parameters are the number of histogram bins, and the pixel and line increment used for sampling when the input image histogram is computed.



The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

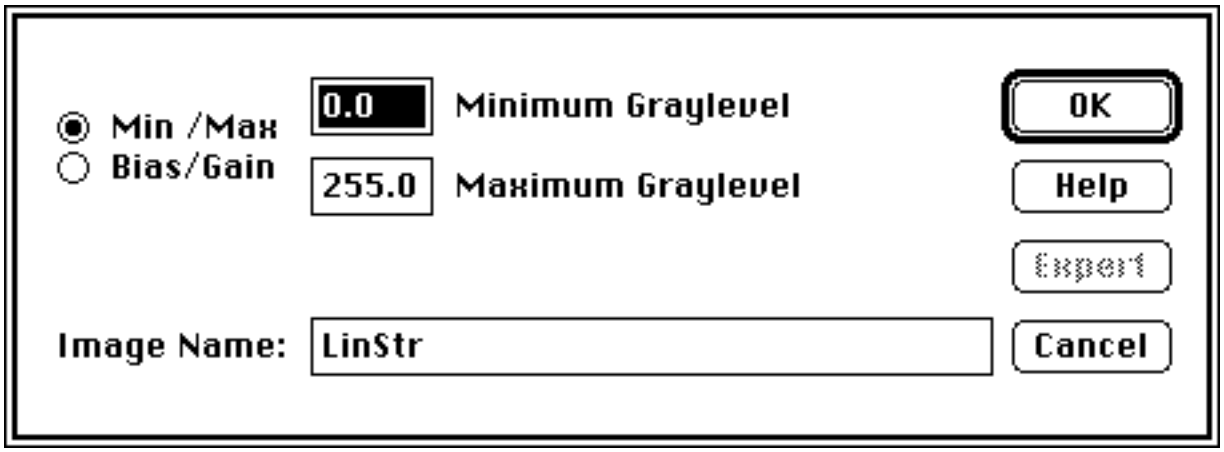
$$O_{ijk} = \mathrm{CDF}\left((b-1) \cdot \frac{I_{ijk} - I_{min}}{I_{max} - I_{min}} + I_{min}\right)$$

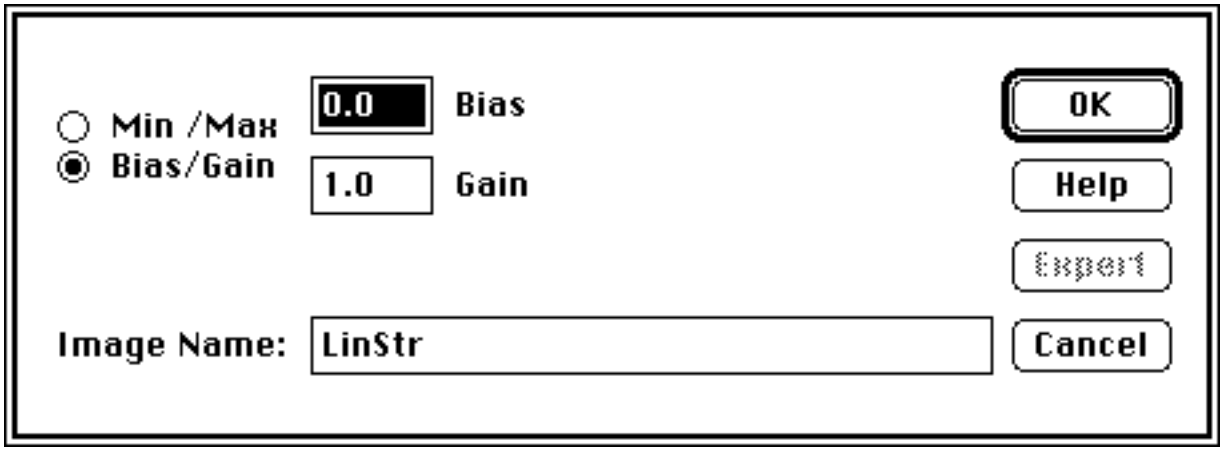
where $\mathrm{CDF}(g) = (I_{max} - I_{min}) \cdot \dfrac{\left|\{I_{ijk} : I_{ijk} \le g\}\right|}{m \cdot n \cdot o}$.

CDF() represents the normalized cumulative distribution function (the normalized cumulative histogram) of the *m*-by-*n*-by-*o* input image *I*, and *b* is the number of histogram bins, as specified in the dialog. As with the Normalize Stretch command, the subsample increment allows you a tradeoff between accuracy and speed. For a typical image with a gaussian histogram, the Histogram Equalize Stretch command tends to reduce image contrast in very light and very dark areas, and to increase contrast in the middle graylevel range.

Linear Stretch

The Linear Stretch command performs a linear graylevel transformation on the input image. The parameters of the stretch may be specified either as the output image's minimum and maximum graylevel values (top dialog), or as the two parameters of a linear coordinate transformation, bias and gain (bottom dialog). You can use the radio buttons to switch between these two equivalent representations. The default parameters are a minimum graylevel of 0 and a maximum graylevel of 255 (resulting in a transformation which makes the image suitable to be stored on disk in 8 bits/pixel format).





The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

$$O_{ijk} = \begin{cases} \dfrac{O_{max} - O_{min}}{I_{max} - I_{min}} \cdot (I_{ijk} - I_{min}) + O_{min} & \text{(if Min/Max are specified)} \\ b + g \cdot I_{ijk} & \text{(if Bias/Gain are specified)} \end{cases}$$

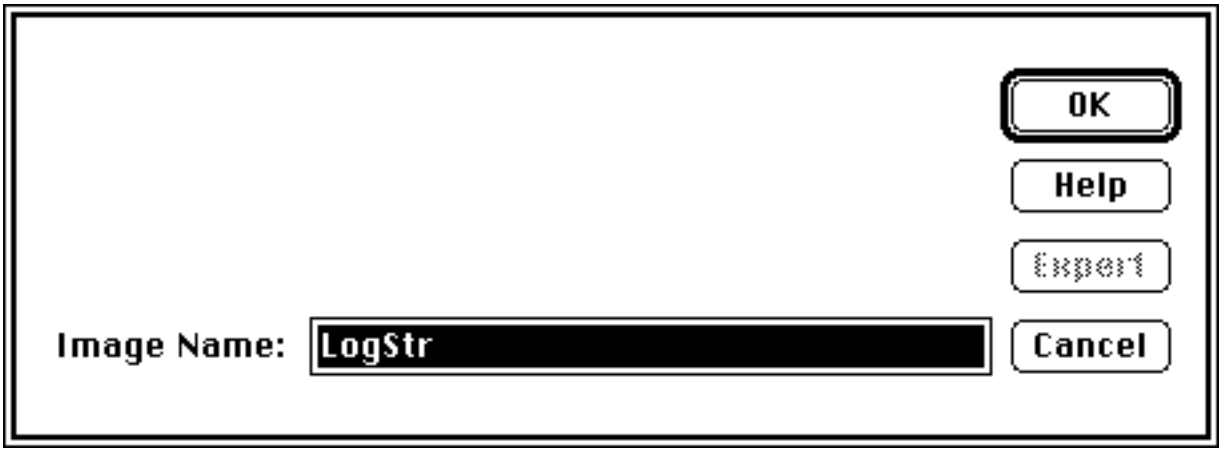
where I_{min} and I_{max} are the minimum and maximum graylevels of the input image, O_{min} and O_{max} are the minimum and maximum graylevels of the output image, and b and g are the graylevel bias and gain, respectively, as specified in the dialog.

If you switch from Min/Max representation to Bias/Gain representation or vice versa, the current parameter settings are transformed accordingly. Note that, due to the automatic display stretch, MacSADIE stretches the graylevel range of each image individually before displaying it, making an image and its linearly stretched version look identical on the screen.

The Linear Stretch command is used to scale image graylevels to some specific range without altering the relative relationship between pixels. This is necessary for commands such as Misc/RGB → HSV, which require image graylevels to be in a certain range. You can also use the Linear Stretch command to create the negative of an image — either by setting the new minimum graylevel to the old maximum and vice versa, or by setting the bias to the old maximum, and the gain to –1.

Logarithmic Stretch

The Logarithmic Stretch command logarithmically stretches the graylevels of the input image.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

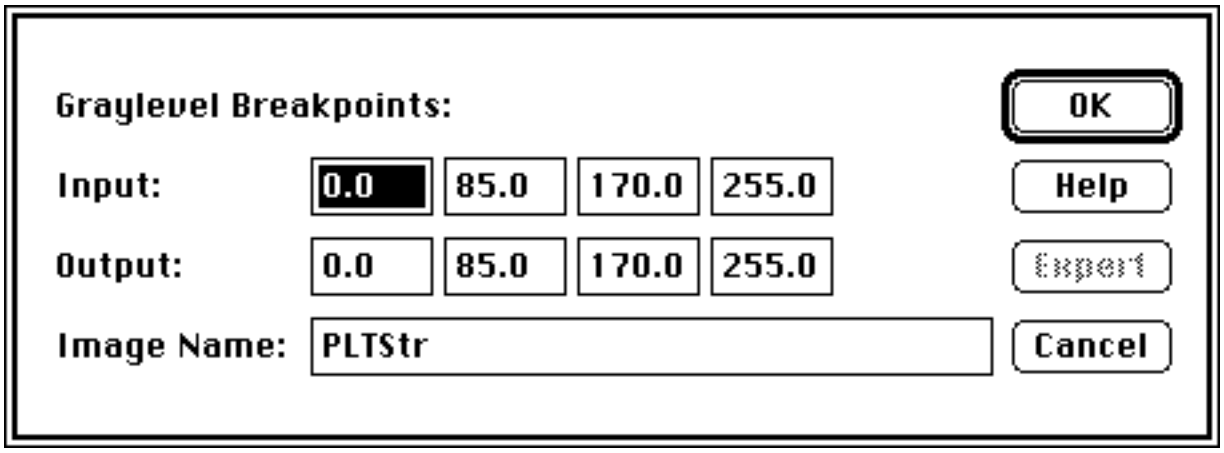
$$O_{ijk} = \begin{cases} \log_{10}(I_{ijk} + 1) & \text{if } I_{ijk} \geq 0 \\ I_{ijk} & \text{if } I_{ijk} < 0 \end{cases}$$

where the function $\log_{10}()$ computes the (base 10) logarithm of its argument.

Since the logarithm of a negative number is undefined, negative graylevels are not transformed, but simply retained. Note that one is added to the original graylevels to allow the transformation of images with graylevel values of zero. The Logarithmic Stretch command is useful for transforming an image with a wide histogram (a very large dynamic graylevel range) skewed towards lower graylevels, such as the images produced by the Analysis/Power Spectrum or Filter/Gradient commands.

Piecewise Linear Stretch

The Piecewise Linear Stretch command stretches image graylevels via a three-segment, piecewise linear transformation. The default graylevel breakpoints in the input image, I_i, $i = 1, 2, 3, 4$, are computed from that image's graylevel minimum and maximum, I_{min} and I_{max}, as $I_i = (i-1) \cdot (I_{max} - I_{min})/3 + I_{min}$.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

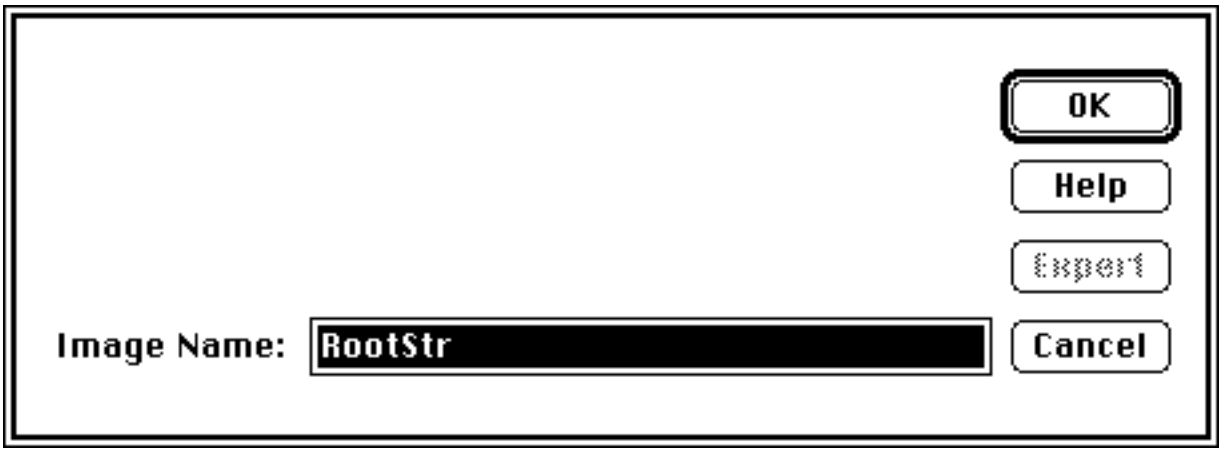
$$O_{ijk} = \begin{cases} \max\{O_1,\ a_1 \cdot I_{ijk} + b_1\} & \text{if } I_{ijk} \le I_2 \\ a_2 \cdot I_{ijk} + b_2 & \text{if } I_2 < I_{ijk} \le I_3 \\ \min\{a_3 \cdot I_{ijk} + b_3,\ O_4\} & \text{if } I_3 < I_{ijk} \end{cases}$$

where $a_i = (O_{i+1} - O_i) / (I_{i+1} - I_i)$ and $b_i = O_i - a_i \cdot I_i$.

The Piecewise Linear Stretch command is useful for images with asymmetric histograms, because it makes it possible to control the amount of stretch applied to different portions of the histogram.

Square Root Stretch

The Square Root Stretch command performs a square root graylevel transformation.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

$$O_{ijk} = \begin{cases} \sqrt{I_{ijk}} & \text{if } I_{ijk} \geq 0 \\ I_{ijk} & \text{if } I_{ijk} < 0 \end{cases}$$

Since the square root of a negative number is undefined, negative graylevels are not transformed, but simply retained. Like the Logarithmic Stretch command, the Square Root Stretch command is useful for reducing the graylevel range of images with a large dynamic range and a histogram skewed towards lower graylevels.

Square Stretch

The Square Stretch command performs a square graylevel transformation.

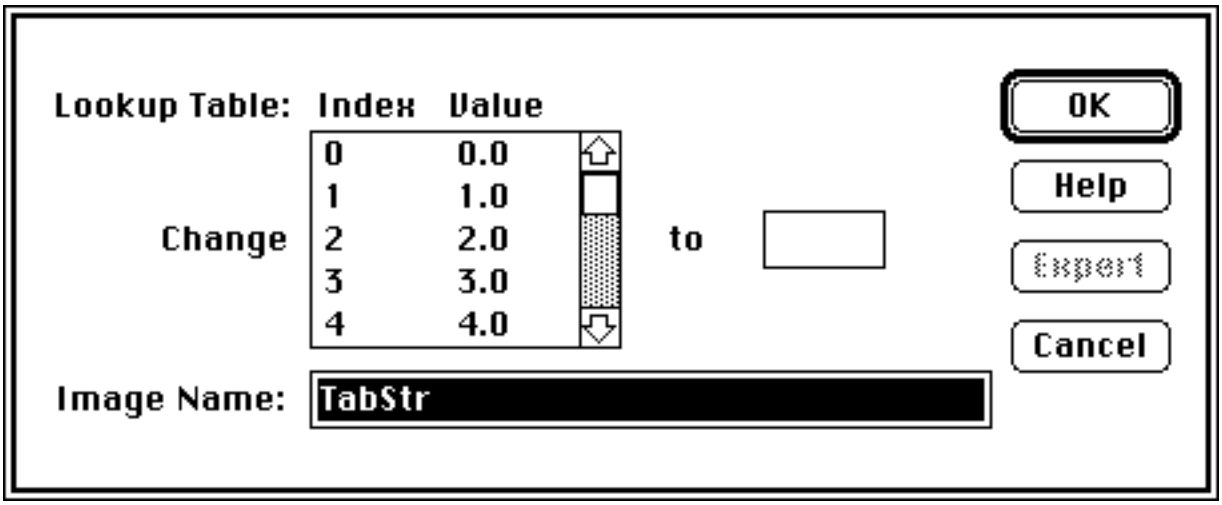
The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

$$O_{ijk} = I_{ijk}^2$$

The Square Stretch command is useful for enlarging the graylevel range of images with a small dynamic range and with a histogram skewed to higher graylevels.

Table Lookup Stretch

The Table Lookup Stretch command performs a graylevel mapping based on a lookup table. The default lookup table for an image is the identity transformation for images with integer graylevel values. To change a lookup table entry, simply select the entry from the scrolling list and modify the value displayed in the edit field at the right.



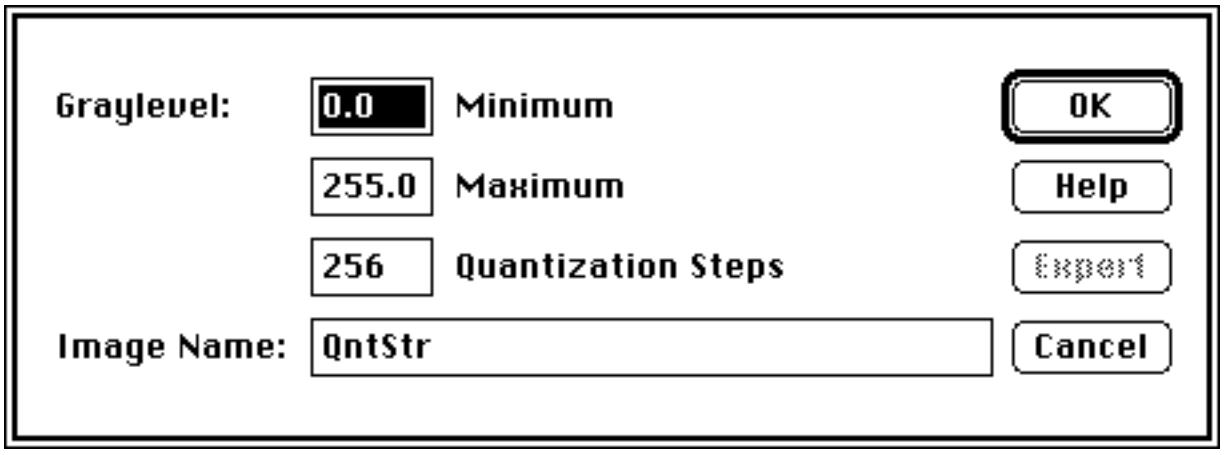
The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

$$O_{ijk} = \mathrm{LUT}(\lfloor I_{ijk} \rfloor)$$

The Table Lookup Stretch command is particularly useful for making specific changes to just a few graylevels in an image, or for relabeling the classes in a classification map. Note that the graylevels are converted to integers before being used to index the lookup table; the Table Lookup Stretch command is therefore not very well suited for images with non-integer graylevel values.

Requantize Graylevels

The Requantize Graylevels command requantizes the input image's graylevels. Parameters are the graylevel values in the input image corresponding to the minimum and maximum graylevels in the output image, and the number of quantization steps in the output image. The minimum and maximum graylevel values default to the input image's graylevel minimum and maximum, respectively. Pixels in the input image with graylevel values outside that range are saturated at the range boundaries.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

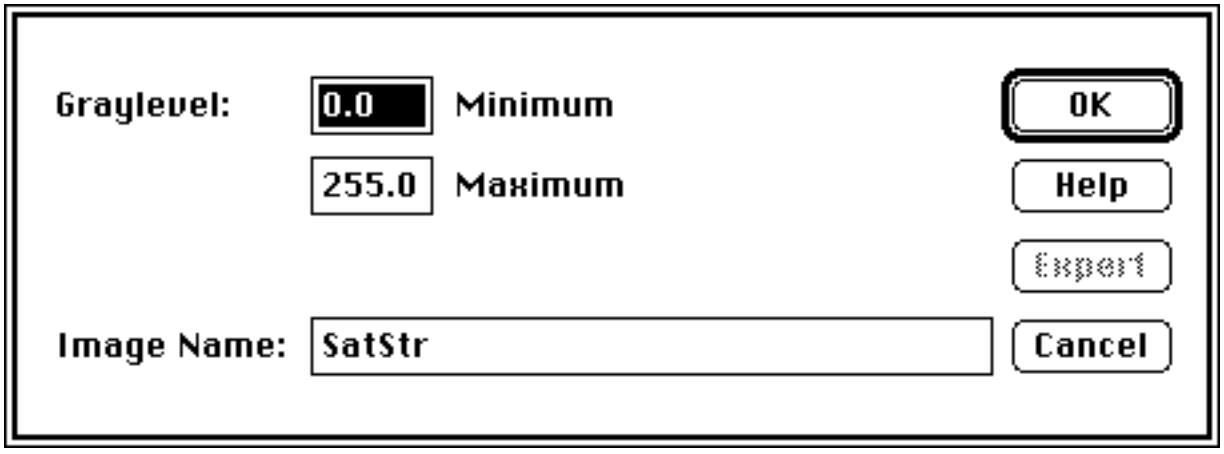
$$O_{ijk} = \max\left\{0, \min\left\{\text{trunc}\left((n-1)\cdot\frac{I_{ijk}-g_{min}}{g_{max}-g_{min}}\right), n-1\right\}\right\}$$

where the function trunc() truncates the fractional part of a real number, n is the number of quantization steps, and g_{min} and g_{max} are the minimum and maximum graylevel values. as specified in the dialog.

Note that g_{min} and g_{max} need not necessarily be identical to the true image graylevel minimum and maximum. The Requantize Graylevels command can be used, for example, for simulating an image acquisition system by setting the parameters to the values of the simulated system.

Saturate Graylevels

The Saturate Graylevels command retains all pixels with graylevel values in a specified range, and saturates all other graylevels at the appropriate range boundary. Parameters are the minimum and maximum graylevel of the range to be retained. They default to the image minimum and maximum (resulting in an identity transformation of the image).



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

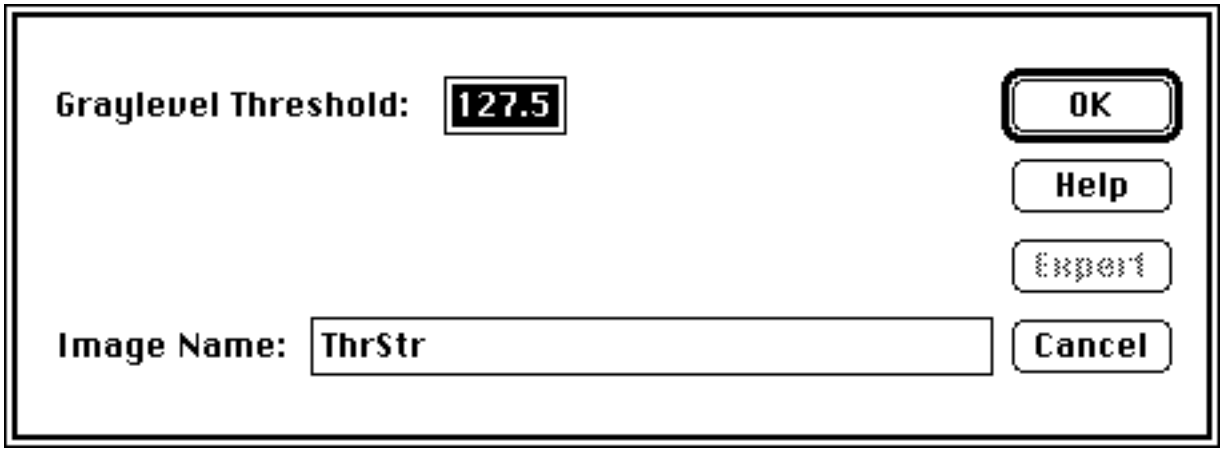
$$O_{ijk} = \begin{cases} g_{min} & \text{if } I_{ijk} < g_{min} \\ I_{ijk} & \text{if } g_{min} \le I_{ijk} \le g_{max} \\ g_{max} & \text{if } I_{ijk} > g_{max} \end{cases}$$

where g_{min} and g_{max} are the minimum and maximum graylevel values to be retained, as specified in the dialog.

The Saturate Graylevels command is useful for discarding (by saturation) extreme graylevel values before further processing, or, in conjunction with MacSADIE's automatic display stretch, for visually stretching the contrast in a displayed image.

Threshold Graylevels

The Threshold Graylevels command converts the input image to a binary image by replacing all graylevel values below some threshold with the minimum graylevel of the input image, I_{min}, and all other graylevel values with the maximum graylevel, I_{max}. The default for the threshold parameter is computed from the minimum and maximum graylevel values as $I_{min} + (I_{min} + I_{max})/2$.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

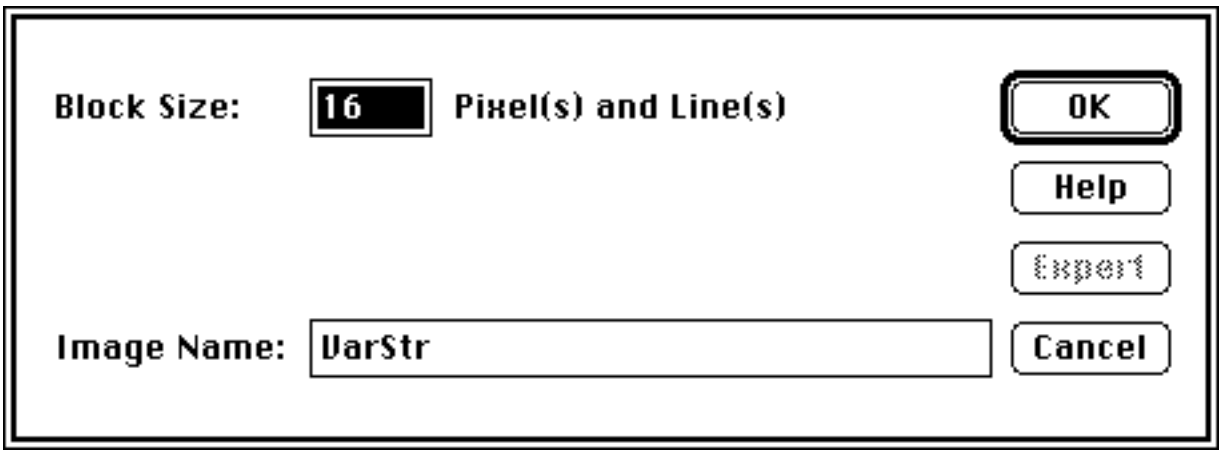
$$O_{ijk} = \begin{cases} I_{min} & \text{if } I_{ijk} < t \\ I_{max} & \text{otherwise} \end{cases}$$

where t is the threshold graylevel specified in the dialog.

The Threshold Graylevels command is useful for simple segmentation and classification tasks.

Adaptive Stretch

The Adaptive Stretch command performs a spatially variable graylevel transformation, where a different linear stretch function is computed for each pixel. The input image (which must be single-band) is partitioned into blocks, and the transformation parameters are calculated for each pixel as a linear interpolation of the stretch parameters for adjoining blocks. The default block size for an *m*-by-*n*-by-*o* image is computed as $\min\{n,o\}/4$.



The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

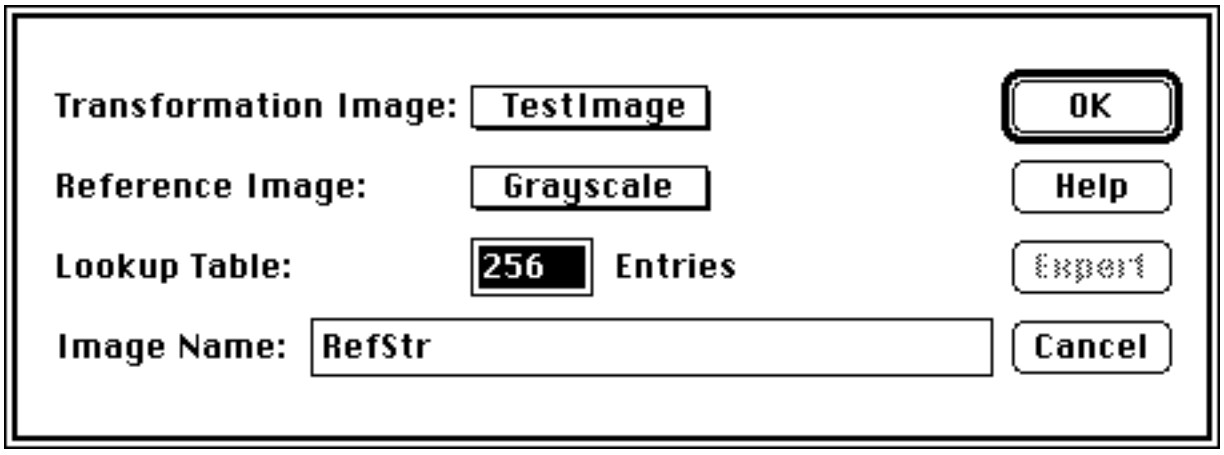
$$O_{jk} = b_{jk} + g_{jk} \cdot I_{jk}$$

where both the bias *b* and the gain *g* depend on the input pixel. For a detailed discussion, including an exact description of the algorithm used for computing the stretch parameters, see J. D. Fahnestock and R. A. Schowengerdt, "Spatially Variant Contrast Enhancement using Local Range Modification," *Optical Engineering*, Vol. 22, No. 3 (May/June 1983), pp. 378–381.

The Adaptive Stretch command is particularly well suited for enhancing the detail in local areas of a scene with widely different mean graylevels or contrasts. Note, however, that the Adaptive Stretch command completely alters the radiometric relationship between pixels, and that the transformation is therefore useful only for visual image enhancement.

Reference Stretch

The Reference Stretch command transforms the graylevels of one image to match, in a least-squares sense, those of another image. This transformation is done via lookup table. Parameters are the image to be transformed (which defaults to the image in the frontmost window), the reference image (it defaults to the image in the window behind it), and the number of entries in the lookup table (this default is computed from the transformation image's minimum and maximum graylevels, I_{min} and I_{max}, as $\lfloor I_{max} \rfloor - \lfloor I_{min} \rfloor + 1$). The two images, which must have the same number of bands, lines/band, and pixels/line, are chosen from pop-up menus of all images currently on the desktop, with all choices permanently enabled.



The graylevel of a pixel in the output image *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

$$O_{ijk} = \mathrm{LUT}\left((n-1)\cdot\frac{I_{ijk}-I_{min}}{I_{max}-I_{min}}+1\right)$$

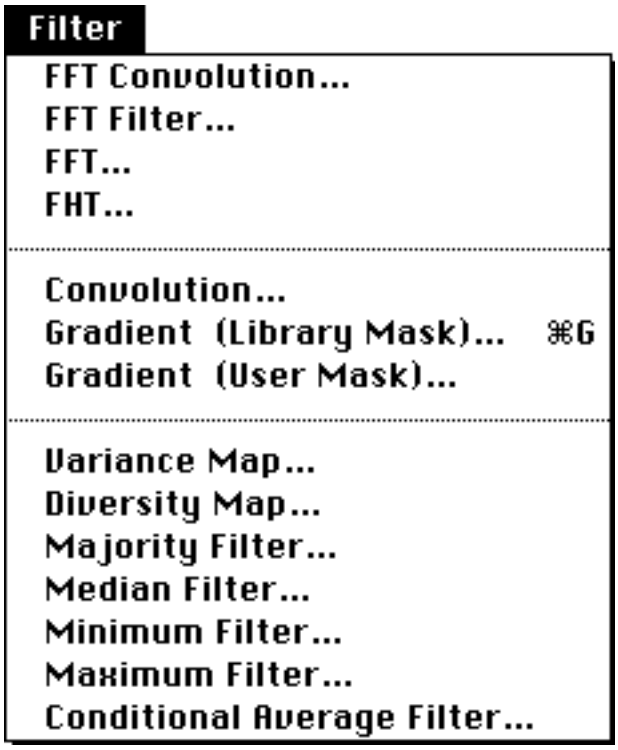
where LUT() is a lookup table, and *n* is the number of lookup table entries specified in the dialog. The exact algorithm for computating the lookup table is described in W. J. Dallas and W. Mauser, "Preparing Pictures for Visual Comparison," *Applied Optics*, Vol. 19, No. 21 (November 1980), pp. 3586–3587.

The Reference Stretch command computes, for each graylevel in the transformation image, the corresponding average graylevel in the reference image. The resulting lookup table is then applied to the transformation image. The reference stretch is used mainly for visual comparison of images that are initially somewhat similar. The Reference Stretch command can produce undesirable results if the two images are dissimilar or uncorrelated.

The Filter Menu

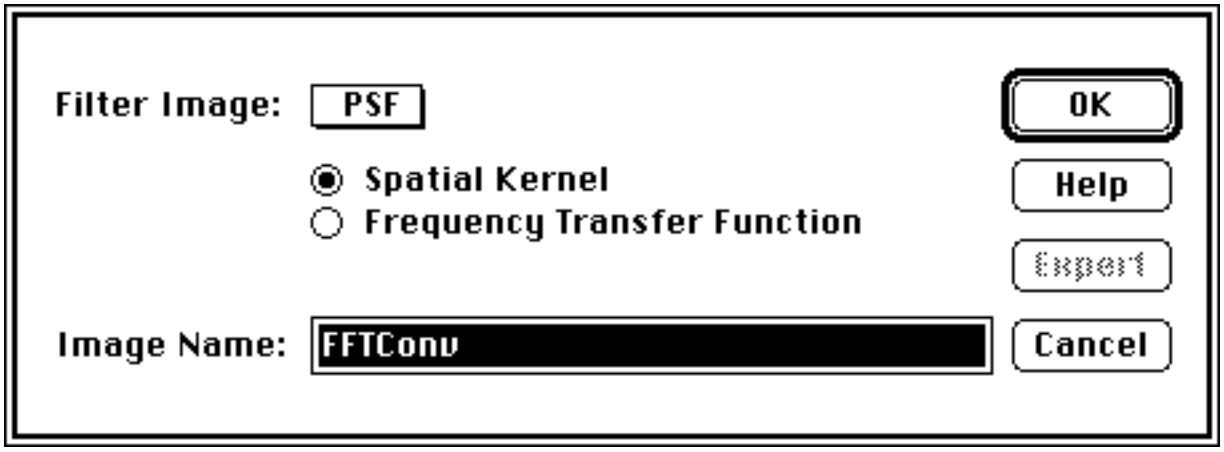
The Filter menu provides commands for filtering images in both the spatial and the frequency domain. Like contrast stretching, filtering is a pixel-by-pixel transformation of the image. Unlike contrast stretching, however, the transformation depends not only on the graylevel of the pixel being processed, but also on the graylevels of neighboring pixels. Consequently, filtering is also termed a neighborhood or context-dependent operation.

With the exception of the Convolution command, all items on the Filter menu are enabled for all types of images. The Convolution command only operates on single-band images, and is enabled only when the frontmost image contains such an image.



FFT Convolution

The FFT Convolution command convolves an image with either a spatial kernel or frequency-domain filter. This is done by computing the discrete Fourier transform of the image, multiplying it either by the Fourier transformed spatial kernel, or by the frequency-domain filter, and computing the inverse Fourier transform of the result. Both the number of lines/band and the number of pixels/line of the input image must be powers of two. The filter image is chosen from a pop-up menu of all images currently on the desktop, with all menu choices permanently enabled. If there is only one image on the desktop, the filter image defaults to that image; otherwise, it defaults to the image in the window behind the frontmost one.



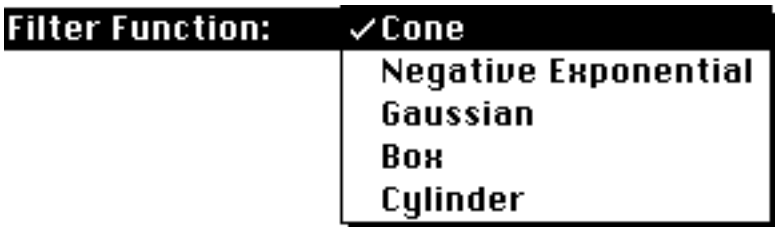
The graylevel of a pixel in the output image *O* is computed from the input image *I* and the filter image *F* as

$$O_{ijk} = \begin{cases} \text{FFT}^{-1}\left(\text{FFT}(I)\cdot\text{FFT}(F)\right)_{ijk} & \text{(if Spatial Kernel is specified)} \\ \text{FFT}^{-1}\left(\text{FFT}(I)\cdot F\right)_{ijk} & \text{(if Frequency Transfer Function is specified)} \end{cases}$$

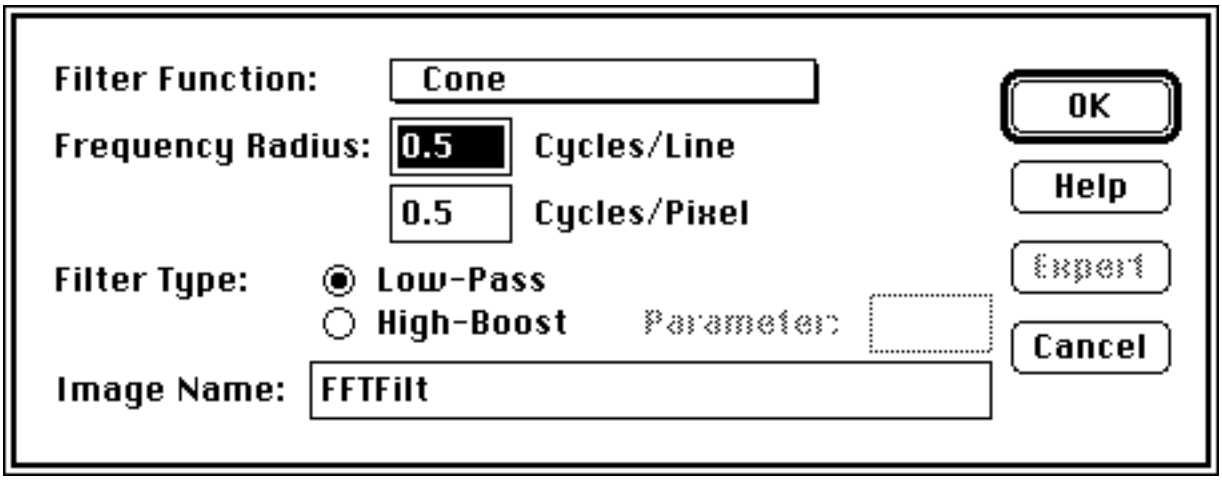
where FFT() denotes the discrete Fourier transform, and FFT^{-1}() the inverse Fourier transform (see the description of the FFT command below for details). The FFT Filter command uses cyclic, wrap-around convolution to determine the pixel values at the image borders.

You can easily create a spatial kernel image with the File/New/Function command. Note that the spatial kernel image may be smaller than the input image — the FFT Convolution command zero-pads the spatial kernel image automatically as required. While producing essentially the same output image as the Convolution command (the only difference is at the image borders due to the cyclic, wrap-around convolution), the FFT Convolution command may be more efficient for large images and/or large kernels. Note that FFT-convolving an image is a memory-intensive operation.

FFT Filter



The FFT Filter command spatially convolves an image by computing the discrete Fourier transform of the image, multiplying it by a specified frequency domain filter function, and computing the inverse Fourier transform of the result. Both the number of lines/band and the number of pixels/line of the input image must be powers of two. The filter function is chosen from a pop-up menu containing a variety of functions, with all menu choices permanently enabled. Other parameters are the radii of the filter function, the filter type, and, in the case of a high-pass filter, a filter parameter. A filter parameter value of one (the default) creates a high-pass filter, and values greater than one result in a parametric high-boost filter.



The graylevel of a pixel in the output image *O* is computed from the input image *I* and the frequency-domain filter *F* as

$$O_{ijk} = \begin{cases} \text{FFT}^{-1}\left(\text{FFT}(I)\cdot F\right)_{ijk} & \text{(if Low-Pass is specified)} \\ \text{FFT}^{-1}\left(\text{FFT}(I)\cdot (c-F)\right)_{ijk} & \text{(if High-Boost is specified)} \end{cases}$$

where *c* is the filter parameter, as specified in the dialog. The FFT Filter command uses cyclic, wrap-around convolution to determine the pixel values at the image borders.

The various filter functions *F* are defined as follows:

Cone: $F_{\xi\zeta} = \begin{cases} (1-r) & \text{if } r \le 1 \\ 0 & \text{otherwise} \end{cases}$

Negative Exponential: $F_{\xi\zeta} = e^{r}$

Gaussian: $F_{\xi\zeta} = e^{-r^2}$

Box: $F_{\xi\zeta} = \begin{cases} 1 & \text{if } f_\xi \le r_\xi \wedge f_\zeta \le r_\zeta \\ 0 & \text{otherwise} \end{cases}$

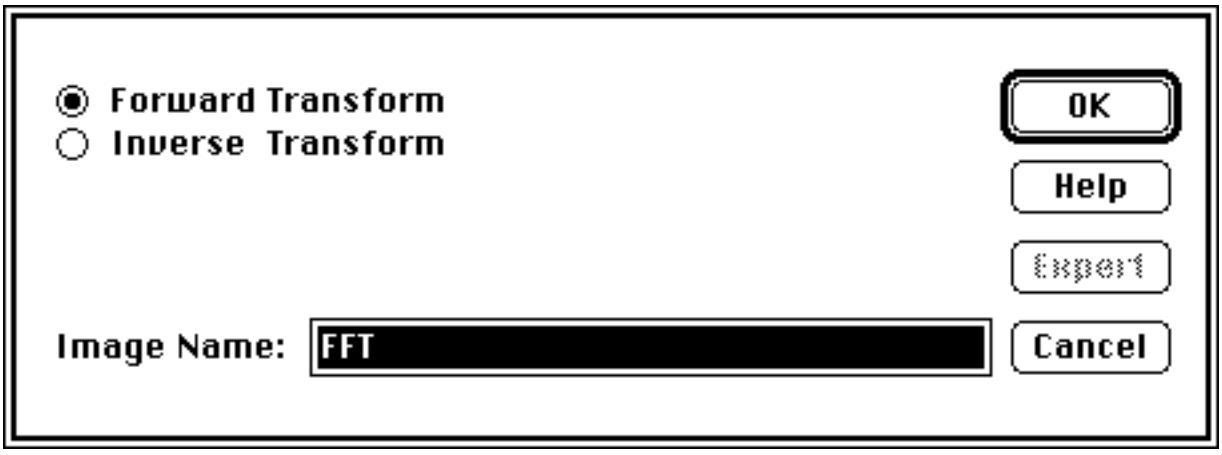
Cylinder: $F_{\xi\zeta} = \begin{cases} 1 & \text{if } r \le 1 \\ 0 & \text{otherwise} \end{cases}$

where $r = \sqrt{(f_\xi / r_\xi)^2 + (f_\zeta / r_\zeta)^2}$, $f_\xi = (0.5 - \xi)/(2\cdot\pi)$ and $f_\zeta = (0.5 - \zeta)/(2\cdot\pi)$ are normalized spatial frequencies, r_ξ and r_ζ are the radii of the filter function, as specified in the dialog, and *e* is the base of the natural logarithm (i.e., $e = \ln(1) = 2.718282\ldots$).

The FFT Filter command is faster than the FFT Convolution command, because only two FFTs are required, compared to three for the FFT Convolution command. All filters available through the FFT Filter command are amplitude-only, and do not modify the phase term. The parametric high-boost filter is discussed in R. A. Schowengerdt, *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York, NY, 1983, pp. 75–79. Generally, the larger the value of the filter parameter, the less the degree of edge enhancement. Note that FFT-filtering an image is a memory-intensive operation.

FFT

The FFT (fast Fourier transform) command performs an image transformation. The forward transform converts the image from the spatial to the frequency domain; the inverse transform converts the image from the frequency to the spatial domain. For the FFT command, both the number of lines/band and the number of pixels/line must be powers of two. **The inverse FFT transform destroys its input image!**



The graylevel of a pixel in the output image *O* is computed from the input image *I* as

$$O_{i\xi\zeta} = \sum_{j=1}^{n} \sum_{k=1}^{o} I_{ijk} \cdot e^{-\iota \cdot (j\xi + k\zeta)} \qquad \text{(if Forward Transform is specified)}$$

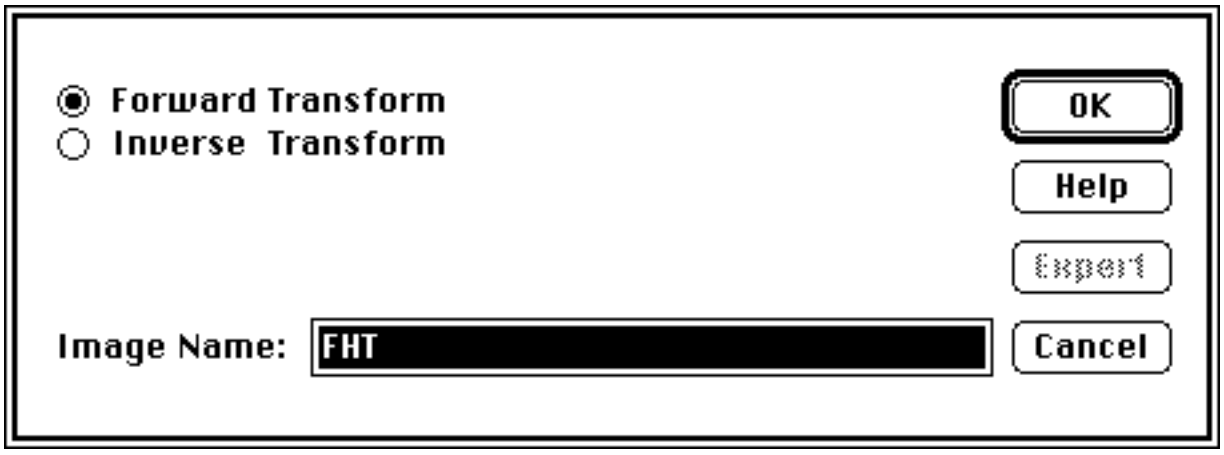
$$O_{ijk} = \sum_{\xi=-\pi}^{\pi} \sum_{\zeta=-\pi}^{\pi} I_{i\xi\zeta} \cdot e^{\iota \cdot (j\xi + k\zeta)} \qquad \text{(if Inverse Transform is specified)}$$

where ξ and ζ are frequency domain coordinates, *e* is the base of the natural logarithm (i.e., $e = \ln(1) = 2.718282\ldots$), and ι is the complex number $\sqrt{-1}$. The spatial domain image (*I* in the case of the forward transform, *O* in the case of the inverse transform) is assumed to be of size *m*-by-*n*-by-*o*. For more information on the fast Fourier transform, see G. D. Bergland, "A guided tour of the fast Fourier transform," *IEEE Spectrum*, Vol. 6, No. 7 (July 1969), pp. 41–52.

Pixel values in the frequency domain are complex numbers (as opposed to graylevels in the spatial domain, which are real numbers). MacSADIE represents complex numbers with two graylevels — one graylevel for the real part, and one for the imaginary part. Consequently, a frequency domain image has twice as many pixels/line as the corresponding spatial domain image, with columns alternating between real and imaginary values. You can separate a complex image into its real and imaginary or magnitude and phase parts with the Misc/Complex → Real command. The FFT command is useful for analyzing the frequency characteristics of an image.

FHT

The FHT (fast Hartley transform) command performs the same transformations between spatial and frequency domains as the FFT command, but uses a different, more efficient algorithm. While MacSADIE's implementation of the FHT algorithm is approximately 30% faster than the FFT implementation, it requires that the number of lines/band equal the number of pixels/line (and that this number be a power of two). The inverse FHT transform does not destroy its input image.



The graylevel of a pixel in the output image *O* is computed from the input image *I* as

$$O_{i\xi\zeta} = \sum_{j=1}^{n} \sum_{k=1}^{o} I_{ijk} \cdot e^{-\iota \cdot (j\xi + k\zeta)} \qquad \text{(if Forward Transform is specified)}$$

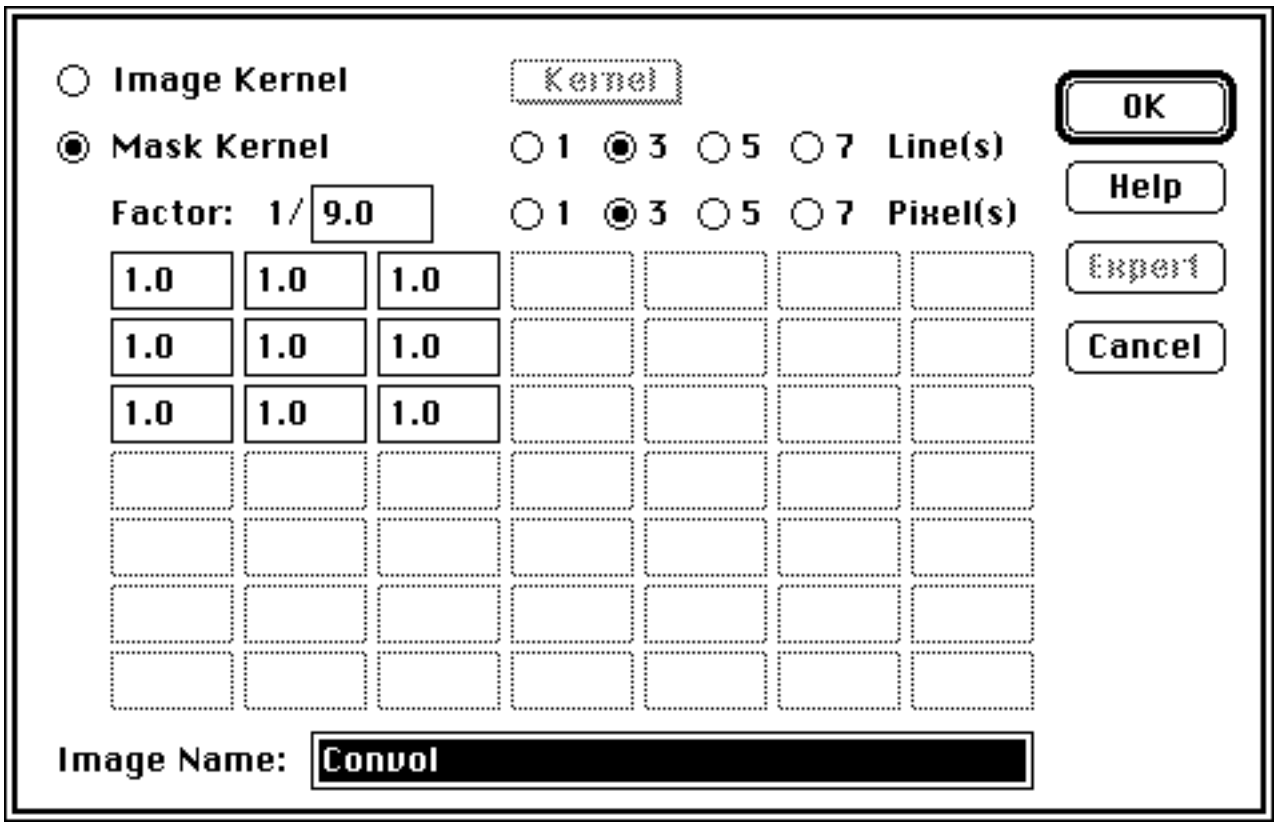
$$O_{ijk} = \sum_{\xi=-\pi}^{\pi} \sum_{\zeta=-\pi}^{\pi} I_{i\xi\zeta} \cdot e^{\iota \cdot (j\xi + k\zeta)} \qquad \text{(if Inverse Transform is specified)}$$

where ξ and ζ are frequency domain coordinates, *e* is the base of the natural logarithm (i.e., $e = \ln(1) = 2.718282\ldots$), and ι is the complex number $\sqrt{-1}$. The spatial domain image (*I* in the case of the forward transform, *O* in the case of the inverse transform) is assumed to be of size *m*-by-*n*-by-*o*. For more information on the fast Hartley transform, see R. N. Bracewell, *The Hartley Transform*, Oxford University Press, New York, NY, 1986.

The format of frequency domain (complex) images is described under the FFT command (see above). The FHT command is useful for analyzing the frequency characteristics of an image, and for efficiently performing convolution and filtering operations on large images.

Convolution

The Convolution command convolves an image with a kernel in the spatial domain. The kernel may be given either as an image, or it may be numerically specified as a mask (the default), up to a maximum size of 7-by-7. If the kernel is specified as a mask (matrix), you need to provide the horizontal and vertical size of the matrix, and the individual matrix values. Each matrix value is multiplied by a factor, which, for a mask of size *m*-by-*n*, defaults to $1/(m \cdot n)$. The factor is adjusted automatically whenever you change the size of the mask. If the spatial kernel is to be given as an image, you can choose from a pop-up menu of all images currently on the desktop, with all menu choices permanently enabled. If there is only one image on the desktop, the spatial kernel defaults to that image; otherwise, it defaults to the image in the window behind the frontmost one. The size of the spatial kernel image must be odd in both the vertical and the horizontal direction, and the spatial kernel image must be single-band. If the spatial kernel is an image, no normalization is performed.



The graylevel of a pixel in the output image O is computed from the input image I and a spatial kernel K of size m-by-n as

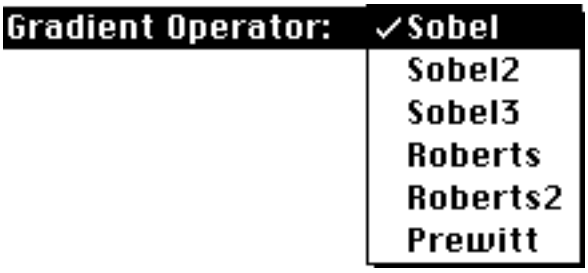
$$O_{jk} = \sum_{p=1}^{m} \sum_{q=1}^{n} I_{j-\left\lceil \frac{m}{2} \right\rceil + p \; k-\left\lceil \frac{n}{2} \right\rceil + q} \cdot K_{pq}$$

Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

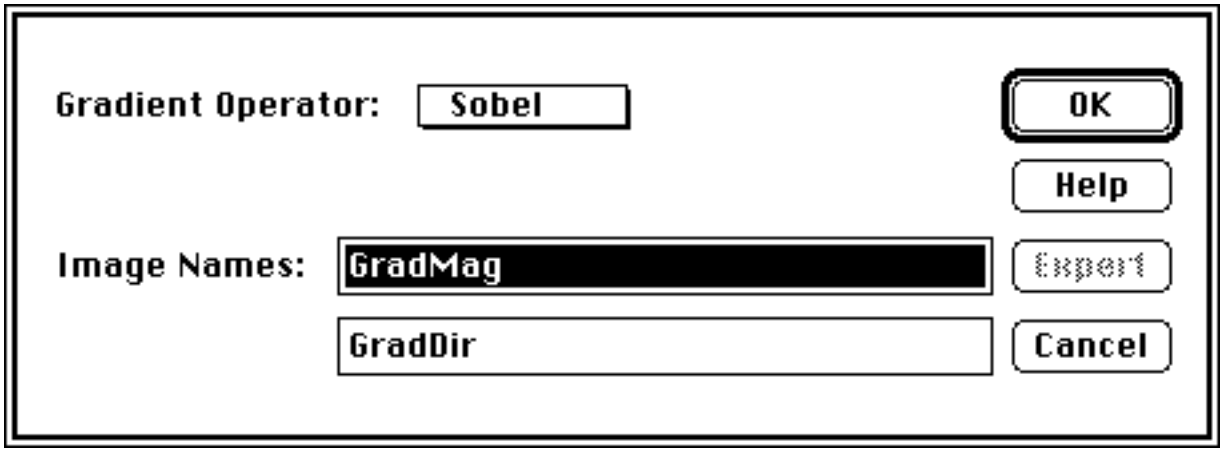
You can easily create a spatial kernel image with the File/New/Function command. You can save a mask kernel as an image for subsequent convolutions as follows. First create a one pixel-by-one pixel wide box function image (File/New/Function with radii of 0.5) at least twice as large as the desired kernel. Then use the Convolution command to filter the box image with the desired kernel. Include an extra factor of 1/255 to normalize the box image to a graylevel of one. Finally, use the Geometry/Window and Subsample command to reduce the filtered image (which is a replication of the mask kernel) to the desired size.

While producing essentially the same output image as the FFT Convolution command (the only difference is at the image borders due to the cyclic, wrap-around convolution used by the FFT Convolution command), the Convolution command may be more efficient when the kernel is small.

Gradient (Library Mask)



The Gradient (Library Mask) command convolves an image with two spatial kernels — a horizontal and a vertical mask. It then combines the results in a vector calculation to produce two output images, one representing the magnitude and the other representing the direction of the gradient at each point of the input image. The masks (gradient operators) are chosen from a pop-up menu of common gradient operators, with all menu choices permanently enabled.



The graylevel of pixels in the magnitude image M and the direction image D are computed from the input image I and two m-by-n kernels K^v and K^h as

$$M_{ijk} = \sqrt{V_{ijk}^2 + H_{ijk}^2}$$

$$D_{ijk} = \tan^{-1}\left(\frac{V_{ijk}}{H_{ijk}}\right)$$

where $V_{ijk} = \sum_{p=1}^{m}\sum_{q=1}^{n} I_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\,k-\left\lceil\frac{n}{2}\right\rceil+q} \cdot K^v_{pq}$, $H_{ijk} = \sum_{p=1}^{m}\sum_{q=1}^{n} I_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\,k-\left\lceil\frac{n}{2}\right\rceil+q} \cdot K^h_{pq}$, v and h denote vertical and horizontal components, respectively, and $\tan^{-1}()$ represents the trigonometric arc tangent function. Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

The following choices are available for the horizontal and vertical kernels, K^h and K^v:

Sobel: $K^h = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$ $K^v = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$

Sobel2: $K^h = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$ $K^v = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$

Sobel3: $K^h = \begin{matrix} 0 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{matrix}$ $K^v = \begin{matrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \end{matrix}$

Roberts: $K^h = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{matrix}$ $K^v = \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{matrix}$

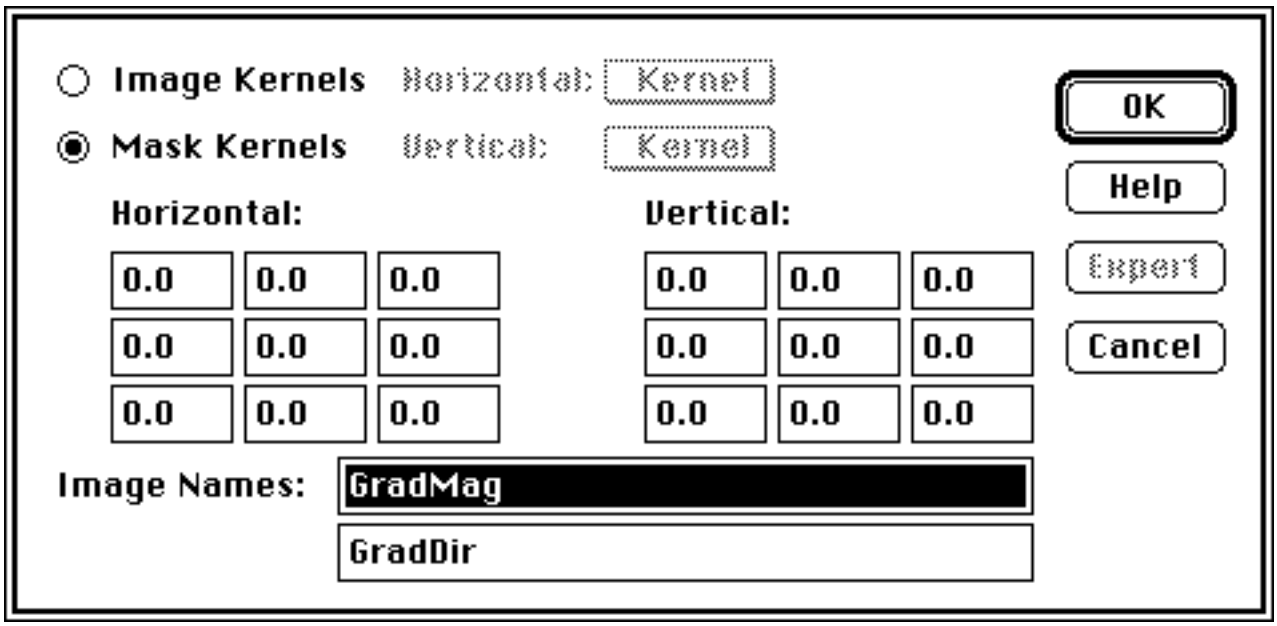
Roberts2: $K^h = \begin{matrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{matrix}$ $K^v = \begin{matrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{matrix}$

Prewitt: $K^h = \begin{matrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{matrix}$ $K^v = \begin{matrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{matrix}$

Gradient operators are useful for detecting edges (i.e., sudden changes in graylevel) in an image. They are discussed in detail in G. S. Robinson, "Detection and Coding of Edges Using Directional Masks," *Optical Engineering*, Vol. 16, No. 6 (November/December 1977), pp. 580–585. Note that the direction map for the Roberts filter is $\pi/4$ radians out of phase with those for the other filters.

Gradient (User Mask)

The Gradient (User Mask) command convolves an image with two spatial kernels — a horizontal and a vertical mask. It then combines the results in a vector calculation to produce two output images, one representing the magnitude, the other representing the direction of the gradient at each point of the input image. The kernels (gradient operators) may be given either as images, or they may be numerically specified as masks (the default). If the kernels are specified as masks (matrices), you need to provide the horizontal and vertical size of the matrices, and the individual matrix values. If the kernels are to be given as images, you can choose from a pop-up menu of all images currently on the desktop, with all menu choices permanently enabled. If there is only one image on the desktop, both kernels defaults to that image; otherwise, they default to the image in the window behind the frontmost one. The size of the kernel images must be odd in both the vertical and the horizontal direction, and the kernel images must be single-band.



The graylevel of pixels in the magnitude image M and the direction image D are computed from the input image I and two m-by-n kernels K^v and K^h as

$$M_{ijk} = \sqrt{V_{ijk}^2 + H_{ijk}^2}$$

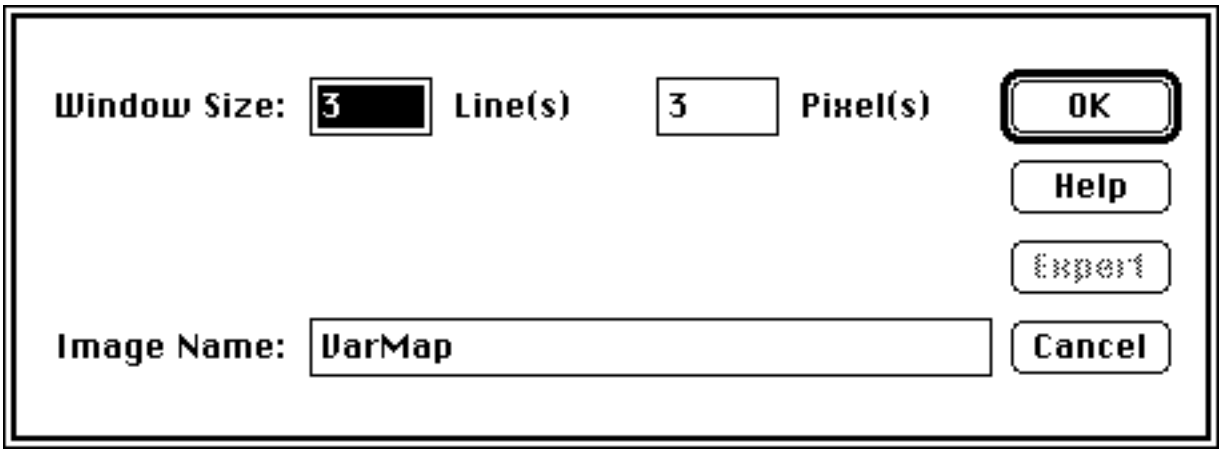
$$D_{ijk} = \tan^{-1}\left(\frac{V_{ijk}}{H_{ijk}}\right)$$

where $V_{ijk} = \sum_{p=1}^{m}\sum_{q=1}^{n} I_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\,k-\left\lceil\frac{n}{2}\right\rceil+q} \cdot K_{pq}^{v}$, $H_{ijk} = \sum_{p=1}^{m}\sum_{q=1}^{n} I_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\,k-\left\lceil\frac{n}{2}\right\rceil+q} \cdot K_{pq}^{h}$, v and h denote vertical and horizontal components, respectively, and $\tan^{-1}()$ represents the trigonometric arc tangent function. Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

Gradient operators are particularly useful for detecting edges (i.e., sudden changes in graylevel) in an image. They are discussed in detail in G. S. Robinson, "Detection and Coding of Edges Using Directional Masks," *Optical Engineering*, Vol. 16, No. 6 (November/December 1977), pp. 580–585. Note that you can use kernels of size 2-by-2 by simply padding a 3-by-3 kernel with zeros.

Variance Map

The Variance Map command computes a map of the input image, where the graylevel of a pixel in the map represents the graylevel variance within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd.



The graylevel of a pixel in the output image *O* is computed from the input image *I* and a window of size *m*-by-*n* as

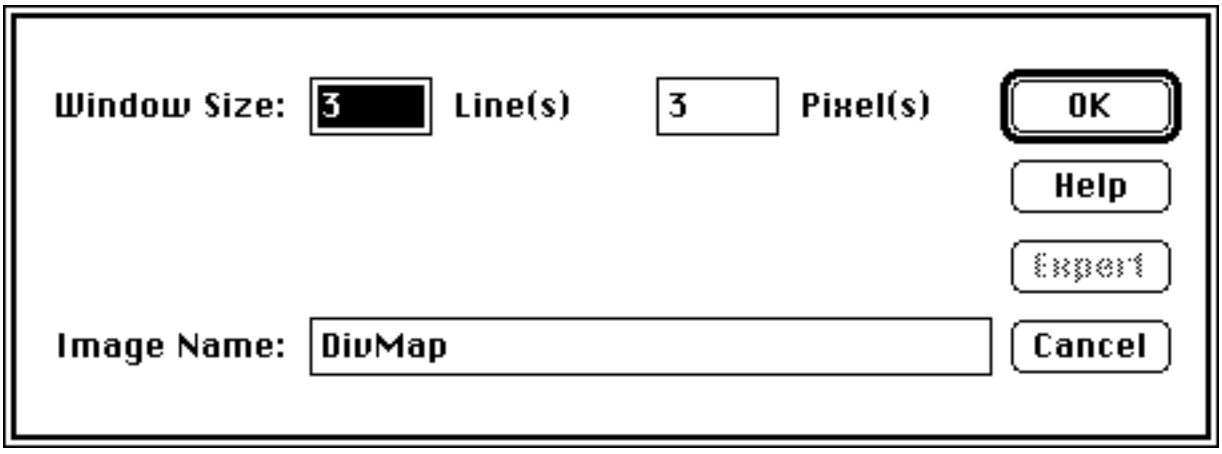
$$O_{ijk} = \frac{\sum_{p=1}^{m}\sum_{q=1}^{n} I^2_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\;k-\left\lceil\frac{n}{2}\right\rceil+q}}{m\cdot n} - \left(\frac{\sum_{p=1}^{m}\sum_{q=1}^{n} I_{i\,j-\left\lceil\frac{m}{2}\right\rceil+p\;k-\left\lceil\frac{n}{2}\right\rceil+q}}{m\cdot n}\right)^2$$

Rows and columns within *m*/2 lines or *n*/2 pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

Like the Diversity Map command (see below), the Variance Map command generates an image representing the local contrast (graylevel variation) in the input image. It is useful for creating a spatial texture or edge activity feature map from an image. With a suitable graylevel threshold applied afterwards (see the Contrast/Threshold Graylevels command), the variance operation can be a useful edge detector for some images.

Diversity Map

The Diversity Map command computes a map of the input image, where the graylevel of a pixel in the map represents the number of different graylevels within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd.



The graylevel of a pixel in the output image *O* is computed from the input image *I* and a window of size *m*-by-*n* as

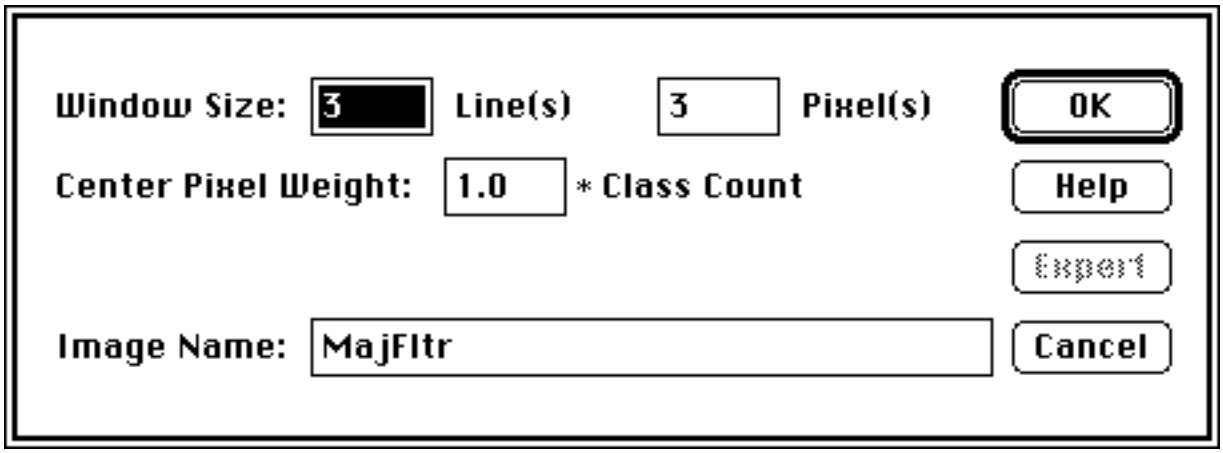
$$O_{ijk} = \left| \left\langle I_{i\,j-\left\lceil \frac{m}{2} \right\rceil + p\;k-\left\lceil \frac{n}{2} \right\rceil + q} : \; 1 \le p \le m \;\wedge\; 1 \le q \le n \right\rangle \right|$$

Rows and columns within *m*/2 lines or *n*/2 pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

The Diversity Map command generates an image representing the local contrast (graylevel variation) in the input image. It is particularly useful for creating a map of local class heterogeneity from a classification map.

Majority Filter

The Majority Filter command filters the input image by creating an output image in which the graylevel of a pixel represents the majority graylevel within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd, and a weight for the influence of the class of the pixel in the center of the window.



The graylevel of a pixel in the output image *O* is computed from the input image *I* and a window of size *m*-by-*n* as

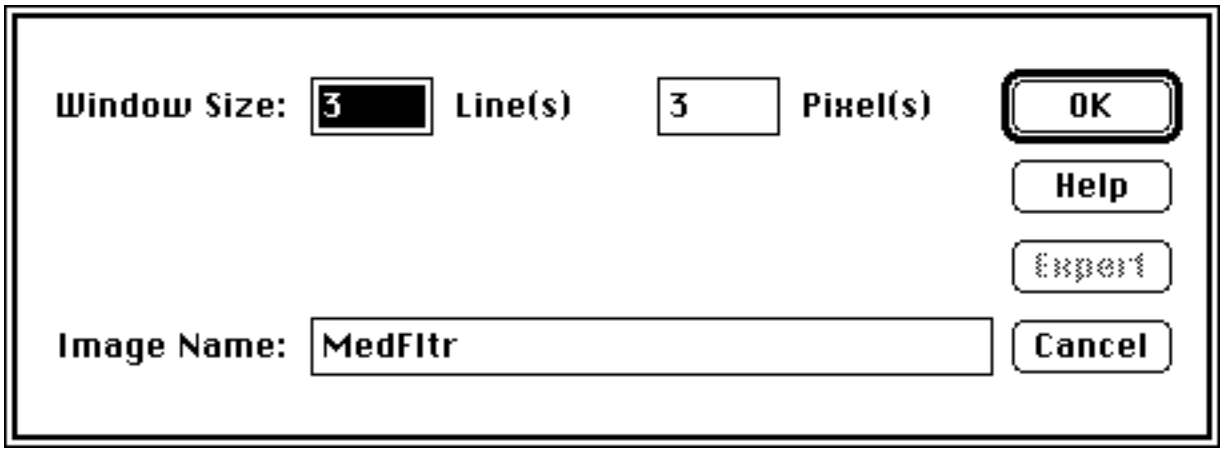
$$O_{ijk} = \begin{cases} I_{ipq} & \text{if } \exists\, p, q : \; f_{pq} > \dfrac{m \cdot n}{2} \;\wedge\; f_{pq} > w \cdot f_{jk} \;\wedge \\ & \qquad j-\left\lceil \frac{m}{2} \right\rceil + 1 \le p \le j-\left\lceil \frac{m}{2} \right\rceil + m \;\wedge\; k-\left\lceil \frac{n}{2} \right\rceil + 1 \le q \le k-\left\lceil \frac{n}{2} \right\rceil + n \\ I_{ijk} & \text{otherwise} \end{cases}$$

where $f_{pq} = \left|\left\langle (r,s):\ I_{i\,j-\left\lceil \frac{m}{2} \right\rceil + r\,k-\left\lceil \frac{n}{2} \right\rceil + s} = I_{ipq} \ \wedge\ 1 \le r \le m \ \wedge\ 1 \le s \le n \right\rangle\right|$, and w is the center pixel weight specified in the dialog. Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column. Pixels within the window are separated into classes according to graylevel values; an input image with integer-valued graylevels is therefore desirable but not required. The majority filter is described in R. A. Schowengerdt, *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York, NY, 1983, pp. 188–190.

Like the Median Filter command and the Conditional Average command (see below), the Majority Filter command is useful for smoothing an image. The amount of smoothing can be controlled to some extent by the center pixel weight parameter. A value of one results in unbiased smoothing, a value less than one biases the smoothing in favor of neighboring pixels, and a value greater than one biases the smoothing in favor of the center pixel. The Majority Filter command is especially useful for classification map smoothing.

Median Filter

The Median Filter command filters an input image by creating an output image where the graylevel of a pixel represents the median graylevel within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd.



The graylevel of a pixel in the output image O is computed from the input image I and a window of size m-by-n as

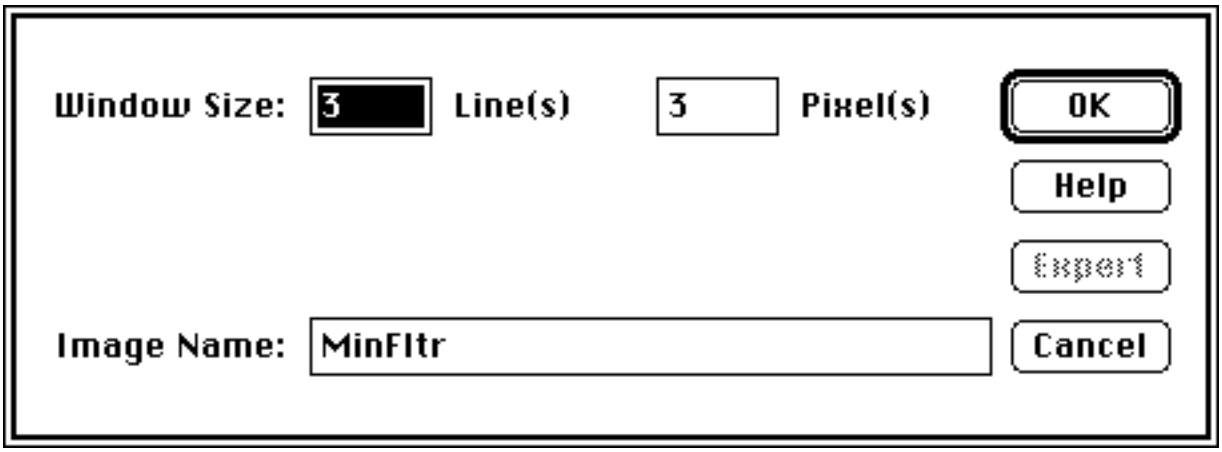
$$O_{ijk} = I_{\left(\left\lceil \frac{m \cdot n}{2} \right\rceil\right)}$$

where $I_{(l)}$, $1 \le l \le m \cdot n$, is a list of the graylevel values of all pixels within the window, in ascending order. Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

The Median Filter command is useful for removing isolated pixel or line noise in an image. For line (or column) noise removal, the window should be rectangular and oriented perpendicular to the noise.

Minimum Filter

The Minimum Filter command filters an input image by creating an output image where the graylevel of a pixel represents the minimum graylevel within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd.



The graylevel of a pixel in the output image O is computed from the input image I and a window of size m-by-n as

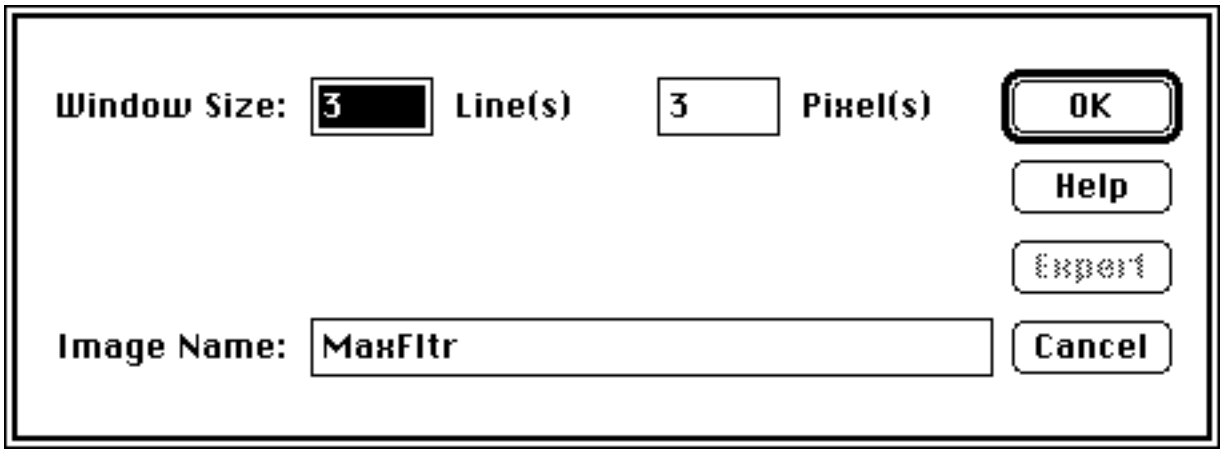
$$O_{ijk} = \min\left\{I_{ipq} : j-\frac{m}{2} \le p \le j+\frac{m}{2} \ \wedge \ k-\frac{n}{2} \le q \le k+\frac{n}{2}\right\}$$

Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

The Minimum command is useful for dilating (expanding) dark objects against a lighter background, or eroding (thinning) light objects against a darker background. In general, the results can not be exactly reversed by using the Maximum Filter command.

Maximum Filter

The Maximum Filter command filters an input image by creating an output image where the graylevel of a pixel represents the maximum graylevel within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd.



The graylevel of a pixel in the output image *O* is computed from the input image *I* and a window of size *m*-by-*n* as

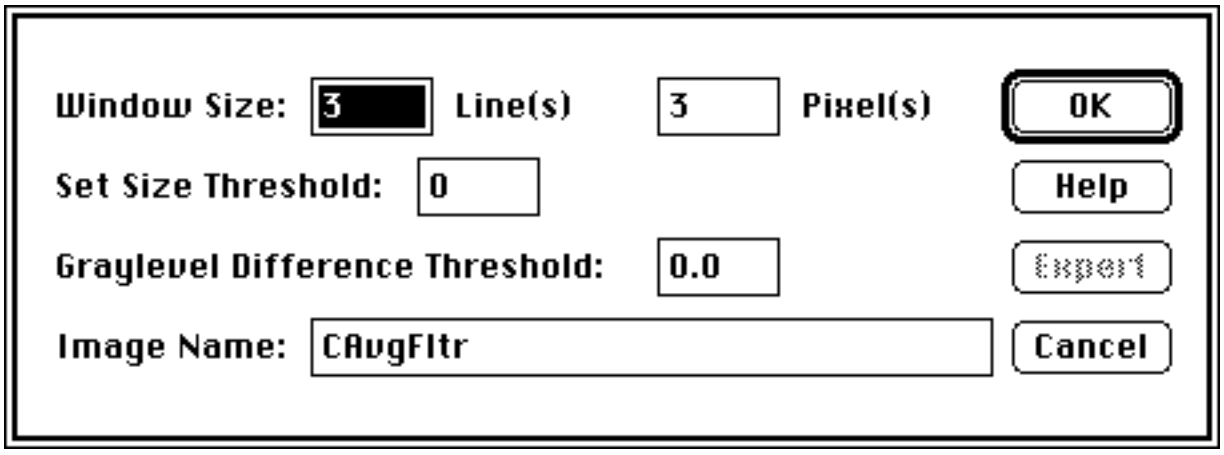
$$O_{ijk} = \max\left\{I_{ipq} : j-\frac{m}{2} \le p \le j+\frac{m}{2} \wedge k-\frac{n}{2} \le q \le k+\frac{n}{2}\right\}$$

Rows and columns within *m*/2 lines or *n*/2 pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column.

The Maximum command is useful for dilating (expanding) light objects against a darker background, or eroding (thinning) dark objects against a lighter background. In general, the results can not be exactly reversed by using the Minimum Filter command.

Conditional Average Filter

The Conditional Average Filter command filters an input image by creating an output image where the graylevel of a pixel represents a conditional average of the graylevels within a window centered at the corresponding pixel of the input image. Parameters are the vertical and horizontal window size, which must both be odd, a set size threshold, and a graylevel difference threshold.



The graylevel of a pixel in the output image O is computed from the input image I and a window of size m-by-n as

$$O_{ijk} = \begin{cases} \frac{1}{|S|} \cdot \sum_{I_{ipq} \in S} I_{ipq} & \text{if } |S| > c \\ \frac{1}{m \cdot n - |S|} \cdot \sum_{I_{ipq} \notin S} I_{ipq} & \text{if } |S| \le c \ \wedge \ \max\left\{ \left| I_{ipq} - I_{irs} \right| : \ I_{ipq} \notin S \ \wedge \ I_{irs} \notin S \right\} < t \\ I_{ijk} & \text{otherwise} \end{cases}$$

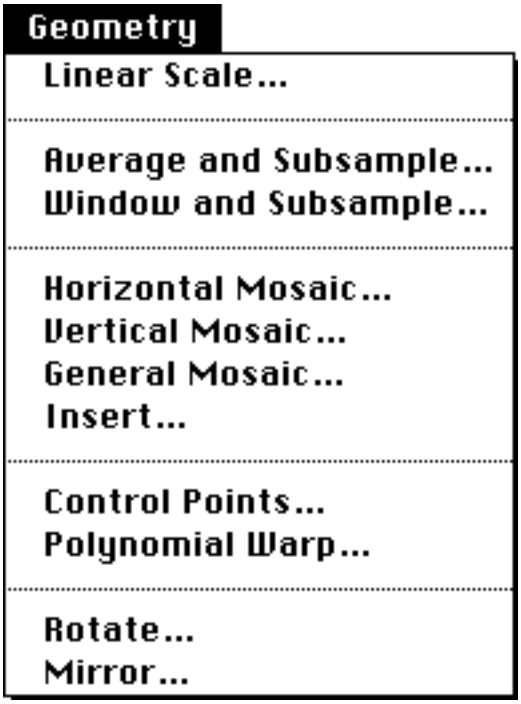
where $S = \left\{ I_{ipq} : \ j - \frac{m}{2} \le p \le j + \frac{m}{2} \ \wedge \ k - \frac{n}{2} \le q \le k + \frac{n}{2} \ \wedge \ |I_{ipq} - I_{ijk}| < t \right\}$, c is the set size (cardinality) threshold, and t is the graylevel difference threshold, as specified in the dialog. Rows and columns within $m/2$ lines or $n/2$ pixels of the borders of the convolved image are filled with graylevel values copied from the nearest valid row or column. This filter is a generalization of the conditional average filter described in T. Ryan, "Image Segmentation Algorithms," *SPIE*, Vol. 534 (1985), pp. 172–178. All pixels within the window whose graylevels differ from that of the center pixel by less than the graylevel threshold t are collected into set S. If the size of S is greater than the cardinality threshold c, the average of the graylevel values in S is assigned as the new graylevel value of the center pixel. If S has cardinality less than or equal to c, and the maximum difference between any two pixels not in the set is less than t, the center pixel is assigned the average graylevel of all pixels not in S. If neither case is true, the center pixel's graylevel value remains unchanged.

Like the Majority Filter command and the Median Filter command (see above), the Conditional Average Filter command is useful for smoothing an image. The first case in the computation above represents the smoothing component of the filter, where the amount of smoothing depends on both the set size threshold and the graylevel difference threshold. The amount of smoothing decreases as the set size threshold increases from 0 to $m \cdot n$, and it increases with an increasing graylevel difference threshold. The second case is useful for eliminating isolated noise in the image, but this aspect of the filter can be suppressed by setting the set size threshold to zero.

The Geometry Menu

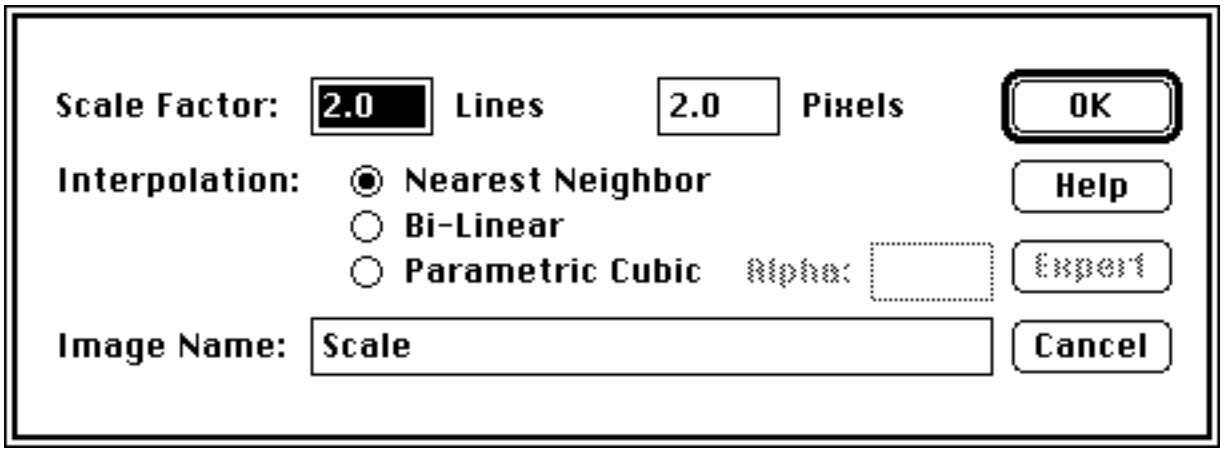
The Geometry menu provides a variety of commands for geometric manipulation, such as scaling or resampling of images, combining images via mosaicing or insertion, computing and applying coordinate transformations, and rotating or mirroring images. These commands are useful for enlarging/reducing images (scaling), simulating image acquisition systems (resampling), adding text or grayscales to existing images (mosaicing or inserting), and registering images to maps or to other images (coordinate transforms).

The Horizontal Mosaic, Vertical Mosaic, General Mosaic, and Insert commands combine two images into a single one, and are therefore enabled only when there are two or more images on the desktop. All other commands are enabled for all types of images.



Linear Scale

The Linear Scale command lets you scale the input image, independently in the horizontal and vertical direction. Parameters are the vertical and horizontal scale factors, the desired interpolation method, and, in the case of parametric cubic interpolation, an interpolation parameter α (which defaults to a value of –0.5).



The graylevel of a pixel in the output image O is computed from the input image I as

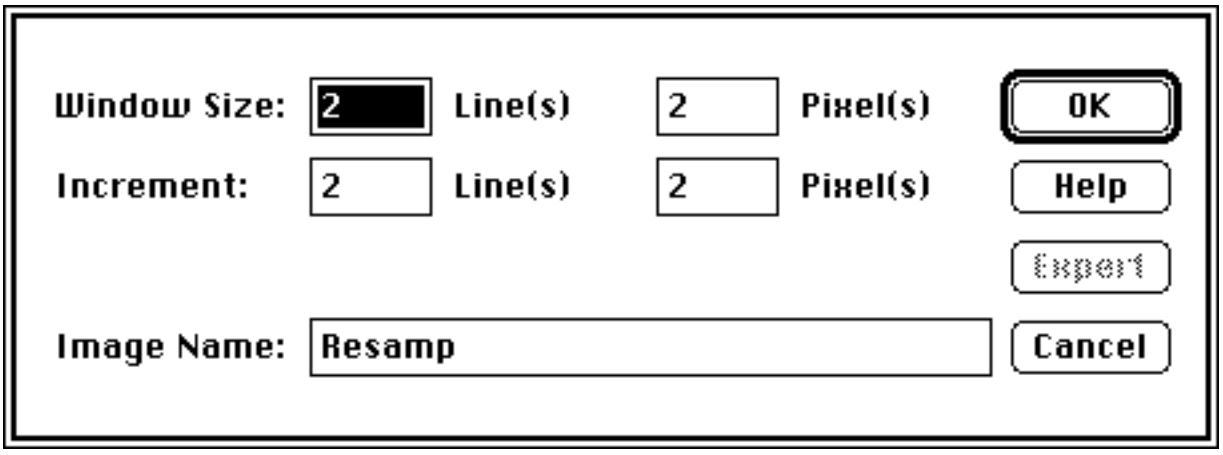
$$O_{ijk} = \begin{cases} \text{nni}(I, i, j/s_j, k/s_k) & \text{(if Nearest Neighbor Interpolation is specified)} \\ \text{bli}(I, i, j/s_j, k/s_k) & \text{(if Bi-Linear Interpolation is specified)} \\ \text{pci}(I, i, j/s_j, k/s_k) & \text{(if Parametric Cubic Interpolation is specified)} \end{cases}$$

where s_j and s_k are the vertical and horizontal scale factors, respectively, as specified in the dialog. For a definition of the interpolation functions nni(), bli(), and pci(), together with a discussion of the advantages and disadvantages of the individual methods, see the description of the Polynomial Warp command (below).

The Linear Scale command is a special case of the more general Polynomial Warp command. It simplifies the parameter input for the frequently occuring case of linear reductions and enlargements, and is considerably faster than the Polynomial Warp command when both scale factors are integers and the nearest neighbor interpolation method is used.

Average and Subsample

The Average and Subsample command resamples the input image by incrementally moving a window over the input image, and averaging the graylevel values of all pixels within the window. Parameters are the vertical and horizontal window size, and the increments for vertical and horizontal movement.



The graylevel of a pixel in the output image O is computed from the input image I as

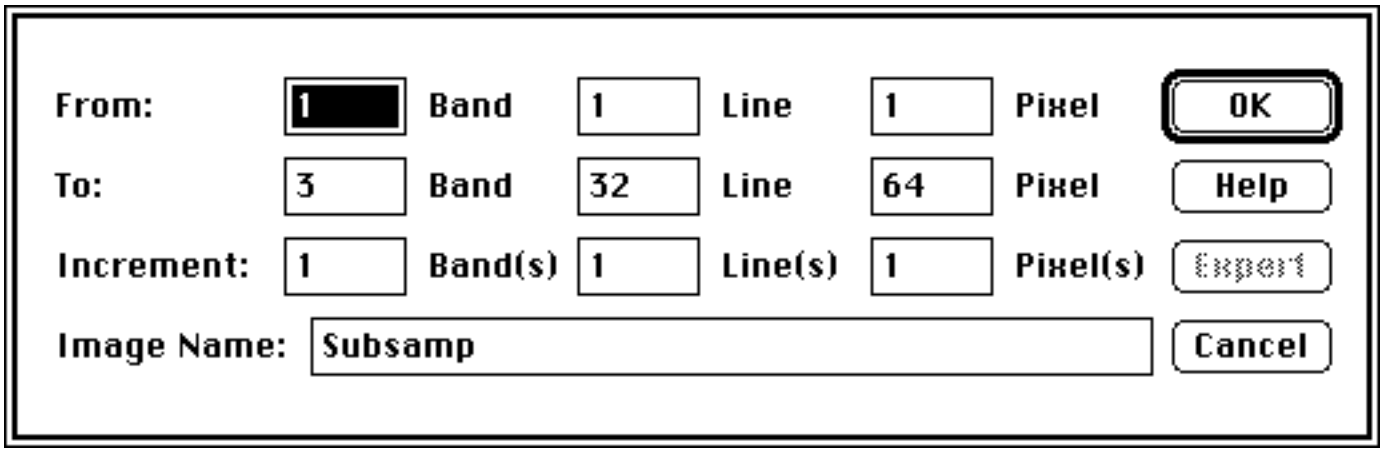
$$O_{ijk} = \frac{\sum_{x=1}^{p} \sum_{y=1}^{q} I_{i\, j \cdot \Delta j + x\;\; k \cdot \Delta k + y}}{p \cdot q}$$

where p and q are the vertical and horizontal window sizes, and Δj and Δk are the line and pixel increments, as specified in the dialog. An input image of size m-by-n-by-o results in an output image of size m-by-$\lceil (n-p)/\Delta k \rceil$-by-$\lceil (o-q)/\Delta k \rceil$.

The Average and Subsample command smooths and reduces the size of the input image simultaneously. The amount of smoothing is dependent on the window size (the larger the window, the more the image is smoothed), while the reduction is directly proportional to the step increment (the larger the increment, the larger the reduction).

Window and Subsample

The Window and Subsample command extracts a window of pixels from the input image, and subsamples it to create the output image. Parameters are the coordinates of the pixels in the window's upper left and lower right corner, and the subsample increments for the band, line, and pixel dimensions.



The graylevel of a pixel in the output image O is computed from the input image I as

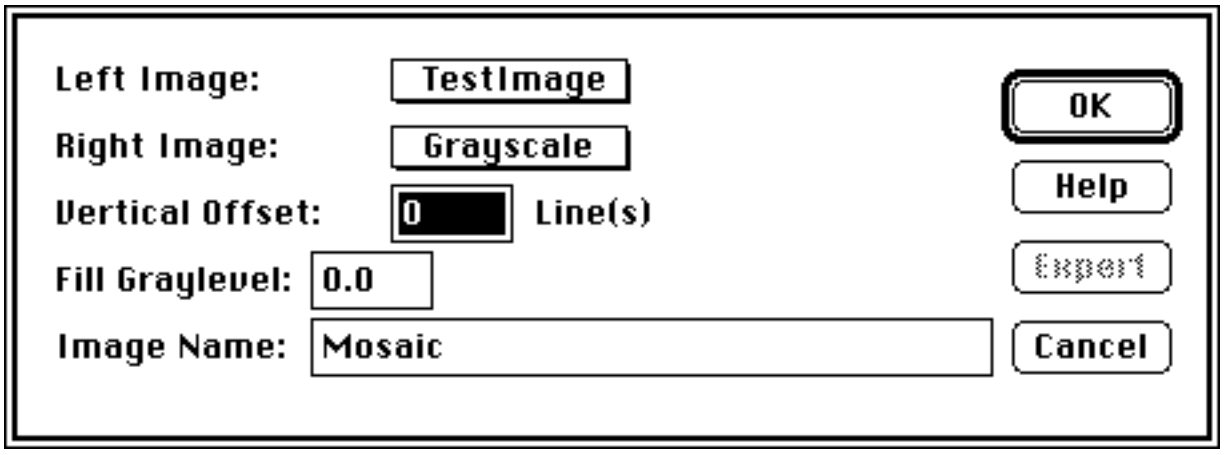
$$O_{ijk} = I_{i \cdot \Delta i + i_{beg} - 1 \; j \cdot \Delta j + j_{beg} - 1 \; k \cdot \Delta k + k_{beg} - 1}$$

where i_{beg}, j_{beg}, and k_{beg} are the coordinates of the first pixel to be subsampled, and Δi, Δj, and Δk are the band, line and pixel increments, respectively, as specified in the dialog. The size of the output image is $\lceil (i_{end} - i_{beg} + 1)/\Delta i \rceil$-by-$\lceil (j_{end} - j_{beg} + 1)/\Delta j \rceil$-by-$\lceil (k_{end} - k_{beg} + 1)/\Delta k \rceil$, for an input image of size m-by-n-by-o.

The Window and Subsample command is useful for precisely extracting (and possibly subsampling) a subimage for further processing. If the size and location of the window to be extracted is not critical, and subsampling is not required, you can use the Edit/Cut and Edit/Paste commands to quickly extract a subimage.

Horizontal Mosaic

The Horizontal Mosaic command concatenates two images horizontally, with the right image vertically offset with respect to the left image. Parameters are the left image (which defaults to the image in the frontmost window), the right image (which defaults to the image in the window behind the frontmost window), a vertical offset between the images (which may be positive or negative), and a graylevel value to be used for the background. The two images, which must have the same number of bands, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled.



The graylevel of a pixel in the output image O is computed from the left input image I and the right input image J as

$$O_{ijk} = \begin{cases} \begin{cases} I_{i\,j-|\Delta j|\,k} & \text{if } 1 \le j-|\Delta j| \le n_I \ \wedge\ k \le o_I \\ J_{i\,j\,k-o_I} & \text{if } 1 \le j \le n_J \ \wedge\ k > o_I \qquad (\text{if } \Delta j < 0) \\ g_{fill} & \text{otherwise} \end{cases} \\ \begin{cases} I_{ijk} & \text{if } 1 \le j \le n_I \ \wedge\ k \le o_I \\ J_{i\,j-|\Delta j|\,k-o_I} & \text{if } 1 \le j-|\Delta j| \le n_J \ \wedge\ k > o_I \qquad (\text{if } \Delta j \ge 0) \\ g_{fill} & \text{otherwise} \end{cases} \end{cases}$$

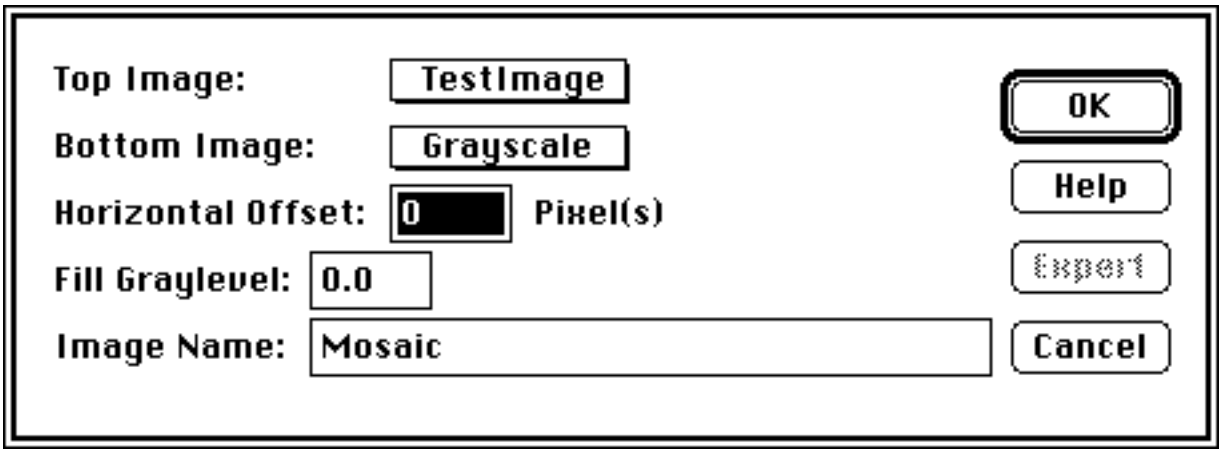
where Δj is the vertical offset, and g_{fill} is the background (fill) graylevel, as specified in the dialog. The output image O is either of size m-by-max$\{n_I+|\Delta j|, n_J\}$-by-(o_I+o_J) (if $\Delta j < 0$), or of size m-by-max$\{n_I, n_J+|\Delta j|\}$-by-(o_I+o_J) (if $\Delta j \ge 0$), for a left image of size m-by-n_I-by-o_I, and a right image of size m-by-n_J-by-o_J.

The Horizontal Mosaic command is useful for appending additional information (such as grayscales or textual annotations) to an image. Another use of the Horizontal Mosaic command is for visually comparing the contrast in multiple images. Due to MacSADIE's automatic display stretch, mosaicing is one way of obtaining a relative comparison of the contrast in multiple images, since the composite image will be displayed with the same graylevel minimum and maximum. The background graylevel will only alter the contrast if it is outside the graylevel range of each one of the mosaiced images — but even then the contrast is altered equally in all images.

Vertical Mosaic

The Vertical Mosaic command concatenates two images vertically, with the top image horizontally offset with respect to the bottom image. Parameters are the top image (which defaults to the image in the frontmost window), the bottom image (which defaults to the image in

the window behind the frontmost window), a horizontal offset between the images (which may be positive or negative), and a graylevel value to be used for the background. The two images, which must have the same number of bands, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled.



The graylevel of a pixel in the output image O is computed from the top input image I and the bottom input image J as

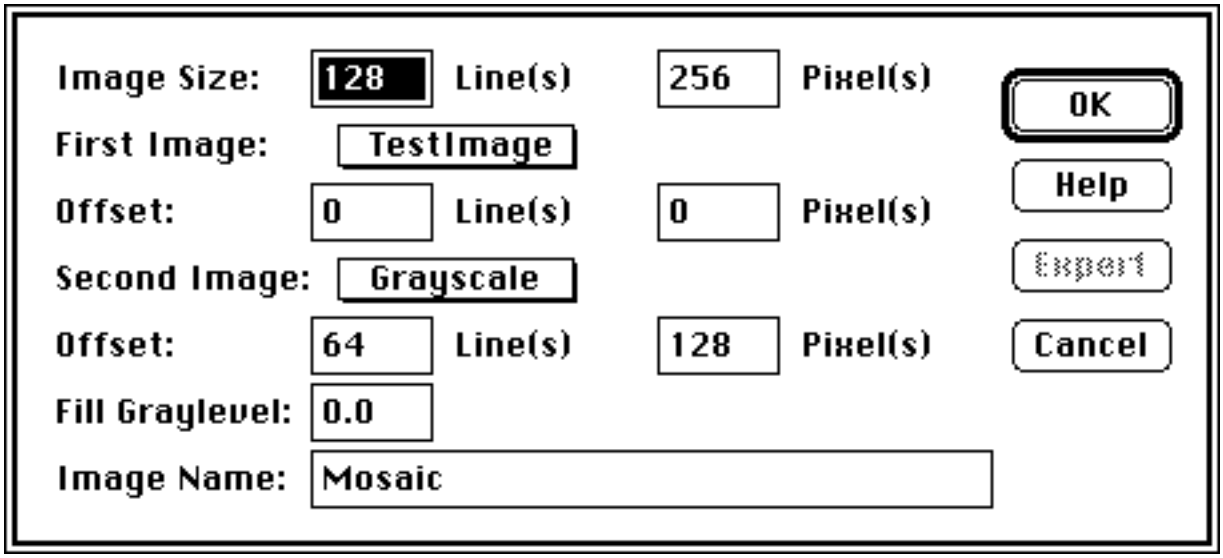
$$O_{ijk} = \begin{cases} \begin{cases} I_{i\,j\,k-|\Delta k|} & \text{if } j \le n_I \wedge 1 \le k-|\Delta k| \le o_I \\ J_{i\,j-n_I\,k} & \text{if } j > n_I \wedge 1 \le k \le o_J \quad (\text{if } \Delta k < 0) \\ g_{fill} & \text{otherwise} \end{cases} \\ \begin{cases} I_{ijk} & \text{if } j \le n_I \wedge 1 \le k \le o_I \\ J_{i\,j-n_I\,k-|\Delta k|} & \text{if } j > n_I \wedge 1 \le k-|\Delta k| \le o_J \quad (\text{if } \Delta k \ge 0) \\ g_{fill} & \text{otherwise} \end{cases} \end{cases}$$

where Δk is the horizontal offset, and g_{fill} is the background (fill) graylevel, as specified in the dialog. The output image O is either of size m-by-(n_I+n_J)-by-max$\{o_I+|\Delta k|, o_J\}$ (if $\Delta k < 0$), or of size m-by-(n_I+n_J)-by-max$\{o_I, o_J+|\Delta k|\}$ (if $\Delta k \ge 0$), for a top image of size m-by-n_I-by-o_I, and a bottom image of size m-by-n_J-by-o_J.

The Vertical Mosaic command is useful for appending additional information (such as grayscales or textual annotations) to an image. Another use of the Vertical Mosaic command is for visually comparing the contrast in multiple images. Due to MacSADIE's automatic display stretch, mosaicing is one way of obtaining a relative comparison of the contrast in multiple images, since the composite image will be displayed with the same graylevel minimum and maximum. The background graylevel will only alter the contrast if it is outside the graylevel range of each one of the mosaiced images — but even then the contrast is altered equally in all images.

General Mosaic

The General Mosaic command lets you compose two images in an arbitrary fashion. The command creates an image of the specified size, fills it with a background graylevel, and places first one and then a second image. Parameters are the size of the output image, the images to be combined, and the vertical and horizontal offsets for each image with respect to the origin of the output image. The default values generate a sufficiently large output image with the first image in the upper left corner, and the lower right corner of the first image touching the upper left corner of the second image. The two images, which must have the same number of bands, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled. The input images must be positioned so that they are located completely inside the output image. If the two input images overlap, the first input image is overwritten.



The graylevel of a pixel in the output image O is computed from the input images I and J as

$$O_{ijk} = \begin{cases} I_{i\,j-\Delta j_I\,k-\Delta k_I} & \text{if } 1 \le j-\Delta j_I \le n_I \;\wedge\; 1 \le k-\Delta k_I \le o_I \;\wedge \\ & \quad \neg\left(1 \le j-\Delta j_J \le n_J \;\wedge\; 1 \le k-\Delta k_J \le o_J\right) \\ J_{i\,j-\Delta j_J\,k-\Delta k_J} & \text{if } 1 \le j-\Delta j_J \le n_J \;\wedge\; 1 \le k-\Delta k_J \le o_J \\ g_{fill} & \text{otherwise} \end{cases}$$

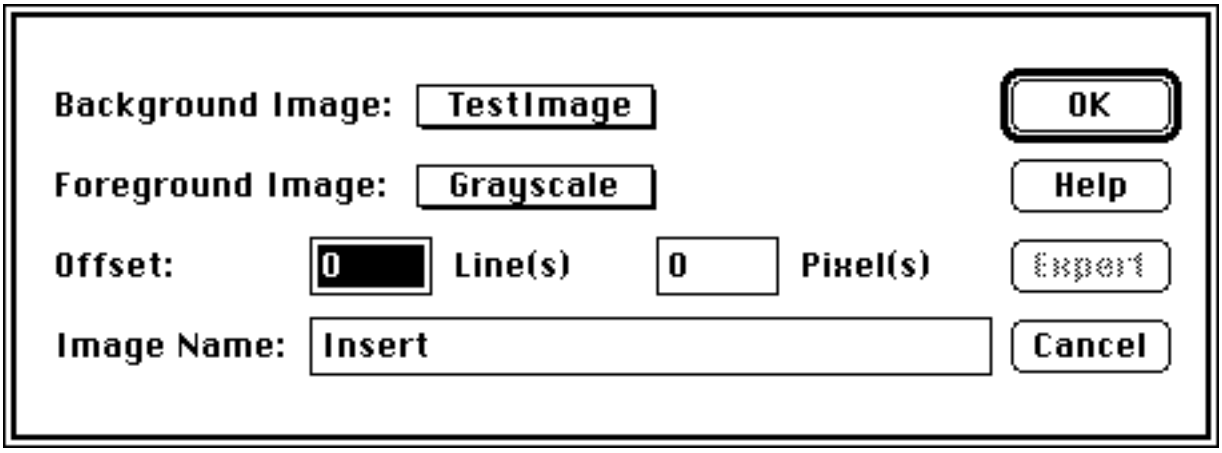
where Δj_I and Δk_I are the vertical and horizontal offsets, respectively, of the first image, Δj_J and Δk_J are the corresponding offsets for the second image, and g_{fill} is the background (fill) graylevel, as specified in the dialog.

The General Mosaic command is useful for appending additional information (such as grayscales or textual annotations) to an image. Another use of the General Mosaic command is for visually comparing the contrast in multiple images. Due to MacSADIE's automatic display

stretch, mosaicing is one way of obtaining a relative comparison of the contrast in multiple images, since the composite image will be displayed with the same graylevel minimum and maximum. The background graylevel will only alter the contrast if it is outside the graylevel range of each one of the mosaiced images — but even then the contrast is altered equally in all images.

Insert

The Insert command lets you overlay one image within another one. Parameters are the background image (which defaults to the image in the frontmost window), the foreground image (which defaults to the image in the window behind the frontmost window), and the offset of the foreground image with respect to the background image. The two images, which must have the same number of bands, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled. The foreground image must be positioned so that it is located completely inside the background image.



The graylevel of a pixel in the output image O is computed from the background image I and the foreground image J as

$$O_{ijk} = \begin{cases} J_{i\,j-\Delta j\,k-\Delta k} & \text{if } 1 \le j-\Delta j \le n_J \wedge 1 \le k-\Delta k \le o_J \\ I_{ijk} & \text{otherwise} \end{cases}$$

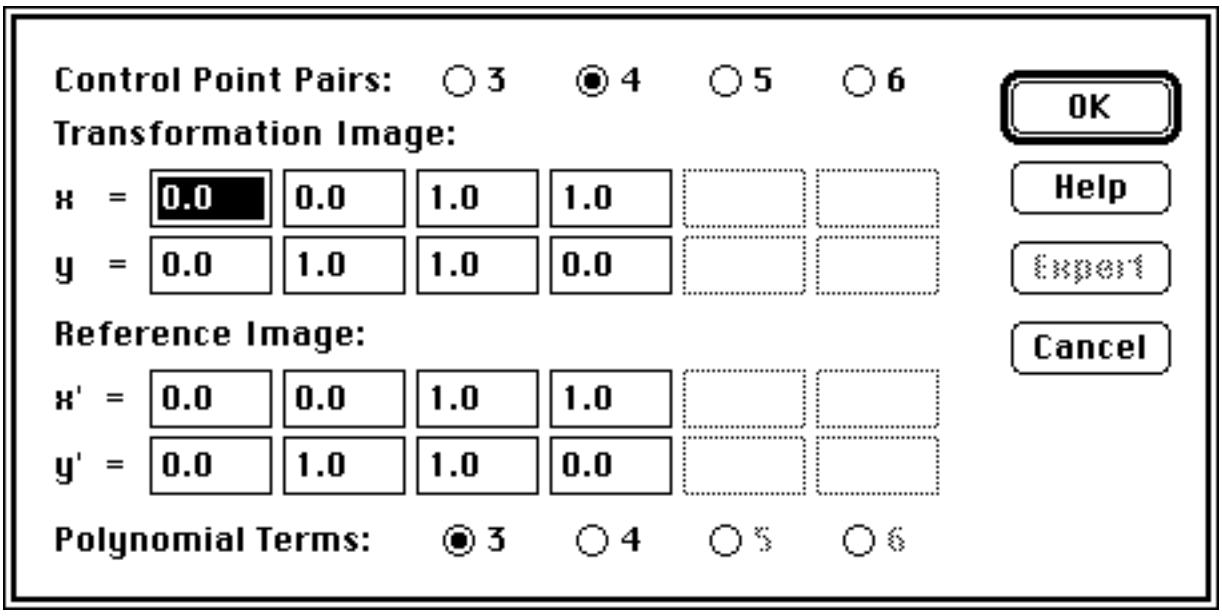
where Δj and Δk are the vertical and horizontal offsets, respectively, as specified in the dialog. For a background image of size m-by-n_I-by-o_I, and a foreground image of size m-by-n_J-by-o_J, the output image is of size m-by-n_I-by-o_I.

The Insert command is useful for inserting additional information (such as grayscales or textual annotations) into an image. It can also be used to create complex test targets, convolution kernels, or frequency filters by combining several simple images created with the File/New commands.

Control Points

The Control Points command computes the polynomial coefficients required to perform a coordinate transformation (“warp”) on an image. The coefficients (polynomial terms) are

computed from the coordinates of a set of control points in both the image to be transformed and a reference image. Parameters are the number of control point pairs, the control point coordinates in both images, and the number of polynomial terms to be computed. The control point coordinates are retained between invocations of the Control Points command. The polynomial coefficients computed by the Control Points command are automatically transferred as defaults to the Polynomial Warp command (see below), which performs the actual coordinate transformation.



Given the coordinates of m control point pairs in the image, $(x_i, y_i,)$ and in a reference coordinate system, (x_i', y_i'), $(1 \le i \le m)$, the Control Points command calculates a least-squares-fit polynomial to the set of control points by solving the following system of linear equations for the transformation coefficients (a_j, b_j), $(1 \le j \le n)$:

$$x_1 = a_1 + a_2 \cdot x_1' + a_3 \cdot y_1' + a_4 \cdot x_1' \cdot y_1' + a_5 \cdot x_1'^2 + a_6 \cdot y_1'^2$$
$$\vdots$$
$$x_m = a_1 + a_2 \cdot x_m' + a_3 \cdot y_m' + a_4 \cdot x_m' \cdot y_m' + a_5 \cdot x_m'^2 + a_6 \cdot y_m'^2$$

$$y_1 = b_1 + b_2 \cdot x_1' + b_3 \cdot y_1' + b_4 \cdot x_1' \cdot y_1' + b_5 \cdot x_1'^2 + b_6 \cdot y_1'^2$$
$$\vdots$$
$$y_m = b_1 + b_2 \cdot x_m' + b_3 \cdot y_m' + b_4 \cdot x_m' \cdot y_m' + b_5 \cdot x_m'^2 + b_6 \cdot y_m'^2$$

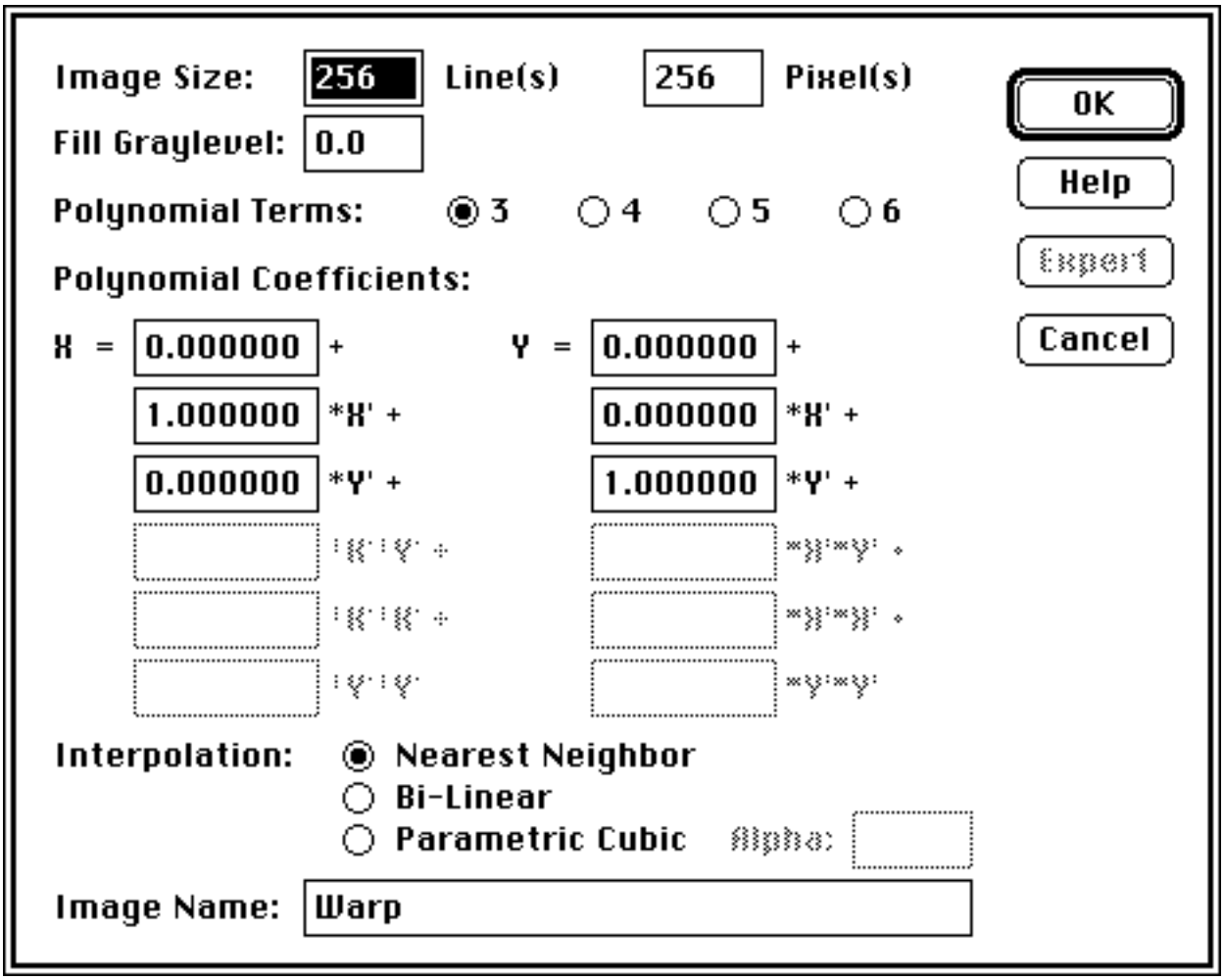
Coordinates x and y are in the coordinate system of the transformation image (the image that will be warped by the transformation), and coordinates x', y' are in the coordinate system of the reference image. The number of control point pairs, m, and the number of polynomial terms, n, are as specified in the dialog.

Solving this system of linear equations requires that there be at least as many control point pairs as there are polynomial terms to be computed (i.e., $m \ge n$). To solve the linear system,

MacSADIE uses a matrix inversion technique. The matrix may become non-invertible (singular), if there are more than two control points lying on a common line (co-linearity condition). It is therefore recommended that the number of control point pairs always be larger than the number of polynomial terms, which also allows distribution of any errors in individual control point coordinates across the whole set of control points (since the linear system is over-determined), as well as the computation of a meaningful "transformation error."

Polynomial Warp

The Polynomial Warp command performs a coordinate transformation on the input image. Parameters are the size of the image to be created, and a graylevel value to be used for the background. The transformation is determined by the number of polynomial terms, the values of the polynomial coefficients, the interpolation method, and, in the case of parametric cubic interpolation, an interpolation parameter α (which defaults to a value of –0.5). The default size of the output image is taken as the size of the input image. The number of polynomial terms and the polynomial coefficients default to the most recent values computed with the Control Points command.



The graylevel of a pixel in the output image O is computed from the input image I as

$$O_{ijk} = \begin{cases} \text{nni}(I,i,y,x) & \text{(if Nearest Neighbor Interpolation is specified)} \\ \text{bli}(I,i,y,x) & \text{(if Bi-Linear Interpolation is specified)} \\ \text{pci}(I,i,y,x) & \text{(if Parametric Cubic Interpolation is specified)} \end{cases}$$

where $x = a_1 + a_2 \cdot k + a_3 \cdot j + a_4 \cdot k \cdot j + a_5 \cdot k^2 + a_6 \cdot j^2,$

$y = b_1 + b_2 \cdot k + b_3 \cdot j + b_4 \cdot k \cdot j + b_5 \cdot b^2 + b_6 \cdot j^2,$

and nni(), bli(), and pci() represent the different interpolation functions, which, for an input image I of size m-by-n-by-o, are defined as follows.

$$\text{nni}(I,i,y,\text{x}) = \begin{cases} I_{i\,\text{rnd}(y)\,\text{rnd}(x)} & \text{if } 1 \le \text{rnd}(y) \le n \ \wedge \ 1 \le \text{rnd}(x) \le o \\ g_{fill} & \text{otherwise} \end{cases}$$

where g_{fill} is the background (fill) graylevel, as specified in the dialog, and the function rnd() rounds its real-valued argument to the nearest integer.

$$\text{bli}(I,i,y,x) = \begin{cases} g_1 + (y - \lfloor y \rfloor) \cdot (g_2 - g_1) & \text{if } 1 \le \lfloor y \rfloor \le n-1 \ \wedge \ 1 \le \lfloor x \rfloor \le o-1 \\ g_{fill} & \text{otherwise} \end{cases}$$

where $g_1 = I_{i\,\lfloor y \rfloor \lfloor x \rfloor} + (x - \lfloor x \rfloor) \cdot (I_{i\,\lfloor y \rfloor \lfloor x \rfloor + 1} - I_{i\,\lfloor y \rfloor \lfloor x \rfloor})$

and $g_2 = I_{i\,\lfloor y \rfloor \lfloor x \rfloor} + (x - \lfloor x \rfloor) \cdot (I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor + 1} - I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor})$

$$\text{pci}(I,i,y,x) = \begin{cases} \text{pwpoly}(\alpha, y - \lfloor y \rfloor, g_1, g_2, g_3, g_4) & \text{if } 2 \le \lfloor y \rfloor \le n-2 \ \wedge \ 2 \le \lfloor x \rfloor \le o-2 \\ g_{fill} & \text{otherwise} \end{cases}$$

where $g_1 = \text{pwpoly}(\alpha, x - \lfloor x \rfloor, I_{i\,\lfloor y \rfloor - 1\,\lfloor x \rfloor - 1}, I_{i\,\lfloor y \rfloor - 1\,\lfloor x \rfloor}, I_{i\,\lfloor y \rfloor - 1\,\lfloor x \rfloor + 1}, I_{i\,\lfloor y \rfloor - 1\,\lfloor x \rfloor + 2}),$

$g_2 = \text{pwpoly}(\alpha, x - \lfloor x \rfloor, I_{i\,\lfloor y \rfloor\,\lfloor x \rfloor - 1}, I_{i\,\lfloor y \rfloor\,\lfloor x \rfloor}, I_{i\,\lfloor y \rfloor\,\lfloor x \rfloor + 1}, I_{i\,\lfloor y \rfloor\,\lfloor x \rfloor + 2}),$

$g_3 = \text{pwpoly}(\alpha, x - \lfloor x \rfloor, I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor - 1}, I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor}, I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor + 1}, I_{i\,\lfloor y \rfloor + 1\,\lfloor x \rfloor + 2}),$

$g_4 = \text{pwpoly}(\alpha, x - \lfloor x \rfloor, I_{i\,\lfloor y \rfloor + 2\,\lfloor x \rfloor - 1}, I_{i\,\lfloor y \rfloor + 2\,\lfloor x \rfloor}, I_{i\,\lfloor y \rfloor + 2\,\lfloor x \rfloor + 1}, I_{i\,\lfloor y \rfloor + 2\,\lfloor x \rfloor + 2}),$

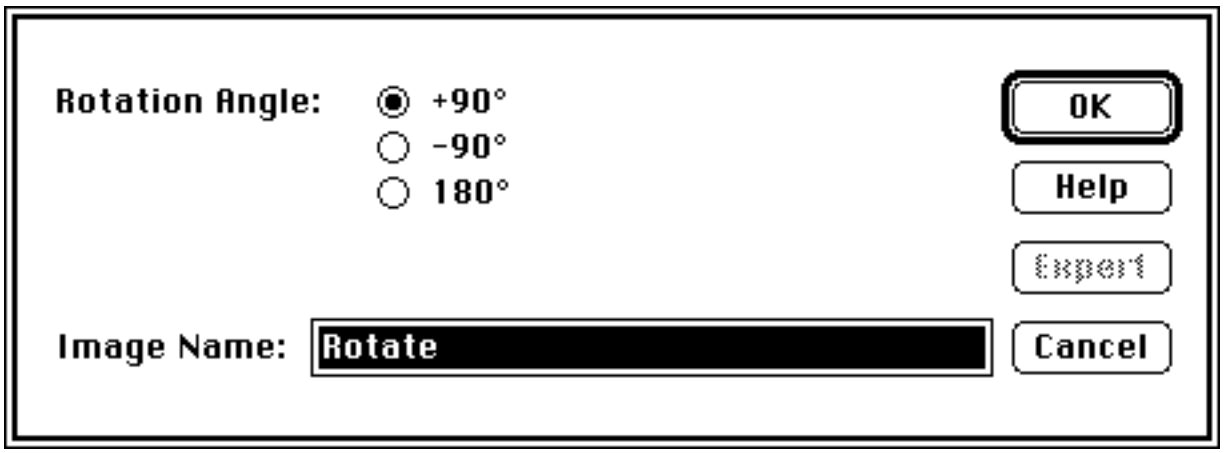
$$\begin{aligned} \text{and } \text{pwpoly}(\alpha, \Delta, e, f, g, h) = {} & \left(-4 + 8 \cdot (1+\Delta) - 5 \cdot (1+\Delta)^2 + (1+\Delta)^3\right) \cdot \alpha \cdot e \ + \\ & \left(1 - (\alpha+3) \cdot \Delta^2 + (\alpha+2) \cdot \Delta^3\right) \cdot f \ + \\ & \left(1 - (\alpha+3) \cdot (1-\Delta)^2 + (\alpha+2) \cdot (1-\Delta)^3\right) \cdot g \ + \\ & \left(-4 + 8 \cdot (2-\Delta) - 5 \cdot (2-\Delta)^2 + (2-\Delta)^3\right) \cdot \alpha \cdot h \end{aligned}$$

The different interpolation methods offer a tradeoff between speed and accuracy. The nearest neighbor interpolation is fastest, whereas the parametric cubic interpolation is generally the most accurate; bi-linear interpolation is a compromise in speed and accuracy. The common cubic interpolation method has $\alpha = -1.0$; however, $\alpha = -0.5$ tends to give superior interpolation results. For a detailed discussion see S. K. Park and R. A. Schowengerdt, "Image Reconstruction by Parametric Cubic Convolution," *Computer Vision, Graphics, and Image Processing*, Vol. 23, No. 3 (September 1983), pp. 258–272.

The Polynomial Warp command is useful for registering an image to another (reference) image or map. Note that the size of the output image should be equal to or greater than that of the reference image, to ensure that all of the common area is included in the warped image.

Rotate

The Rotate command rotates the frontmost image by multiples of 90 degrees.



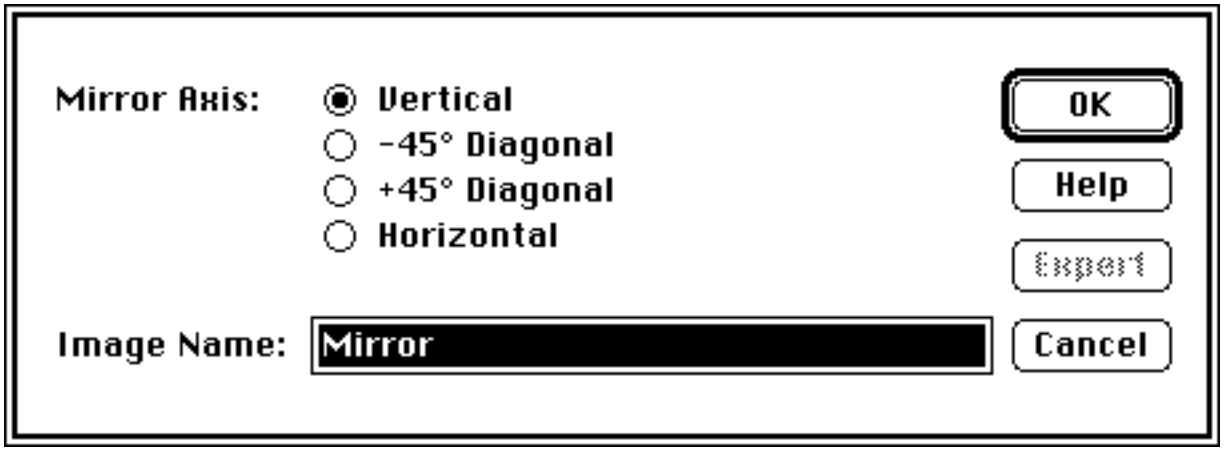
The graylevel of a pixel in the output image O is computed from an m-by-n-by-o input image I as

$$O_{ijk} = \begin{cases} I_{i\ n-k+1\ j} & \text{(if } +90^\circ \text{ is specified)} \\ I_{i\ k\ o-j+1} & \text{(if } -90^\circ \text{ is specified)} \\ I_{i\ n-j+1\ o-k+1} & \text{(if } 180^\circ \text{ is specified)} \end{cases}$$

where the output image is of size m-by-n-by-o if the input image is rotated by 180˚, and of size m-by-o-by-n otherwise.

Mirror

The Mirror command mirrors the frontmost image along one of four axes.



The graylevel of a pixel in the output image O is computed from an m-by-n-by-o input image I as

$$O_{ijk} = \begin{cases} I_{i\ j\ n-k+1} & \text{(if Vertical is specified)} \\ I_{i\ k\ j} & \text{(if } -45^\circ \text{ Diagonal is specified)} \\ I_{i\ o-k+1\ n-j+1} & \text{(if } +45^\circ \text{ Diagonal is specified)} \\ I_{i\ n-j+1\ k} & \text{(if Horizontal is specified)} \end{cases}$$

where the output image is of size m-by-n-by-o if the input image is mirrored along the vertical or along the horizontal axis, and of size m-by-o-by-n otherwise.

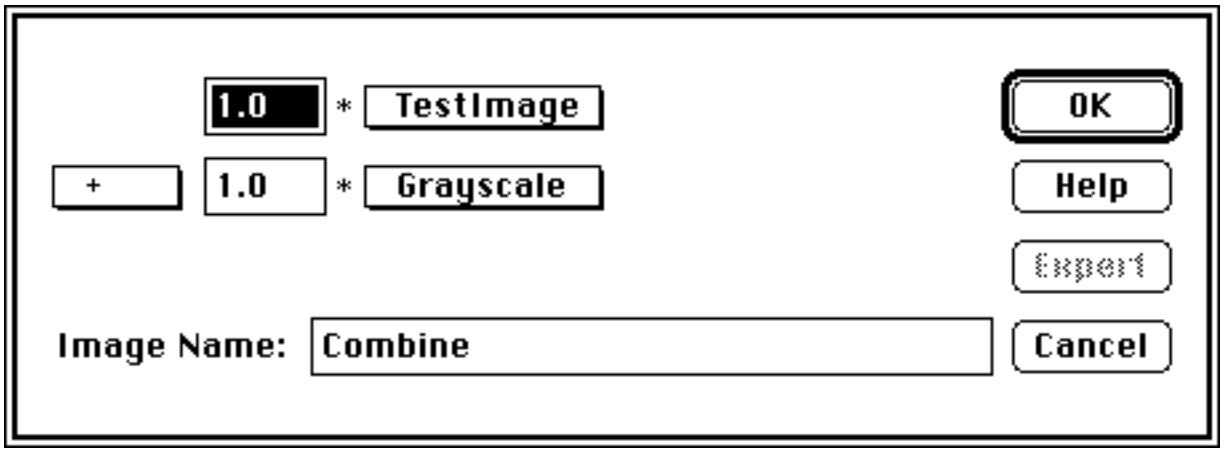
The Miscellaneous Menu

The Miscellaneous menu provides a variety of useful commands for composing images, transforming images from one domain or format into another, and for adding gridlines and hashmarks to an image.

The Combine and Compare commands compose two images into a single one and are enabled only when there are at least two images on the desktop. The Principal Components Transform command operates on multi-band images; it is enabled only when the image in the frontmost window contains a multi-band image. The RGB → HSV, HSV → RGB, and BIL → BSQ commands require three-band (color) images; they are enabled only when the image in the frontmost window is a three-band image. The BSQ → BIL, Complex → Real, Real → Complex, and Grid commands are enabled for all types of images.

Combine

The Combine command computes a weighted linear combination of two images. Parameters are the two images, their respective weights, and the combination function. The first image defaults to the image in the frontmost window, the second image to the one directly behind it. The two images, which must have the same number of bands, lines/band, and pixels/line, are chosen from pop-up menus of all images currently on the desktop. The combination function is chosen from a pop-up menu, with all menu choices permanently enabled.



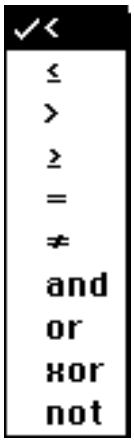
The graylevel of a pixel in the output image O is computed from the graylevels of the corresponding pixels in the input images I and J as

$$O_{ijk} = \begin{cases} (w_I \cdot I_{ijk}) + (w_J \cdot J_{ijk}) & \text{(if "+" is specified)} \\ (w_I \cdot I_{ijk}) - (w_J \cdot J_{ijk}) & \text{(if "–" is specified)} \\ (w_I \cdot I_{ijk}) \cdot (w_J \cdot J_{ijk}) & \text{(if "*" is specified)} \\ (w_I \cdot I_{ijk}) / (w_J \cdot J_{ijk}) & \text{(if "/" is specified)} \\ \min\{w_I \cdot I_{ijk}, w_J \cdot J_{ijk}\} & \text{(if "min" is specified)} \\ \max\{w_I \cdot I_{ijk}, w_J \cdot J_{ijk}\} & \text{(if "max" is specified)} \end{cases}$$

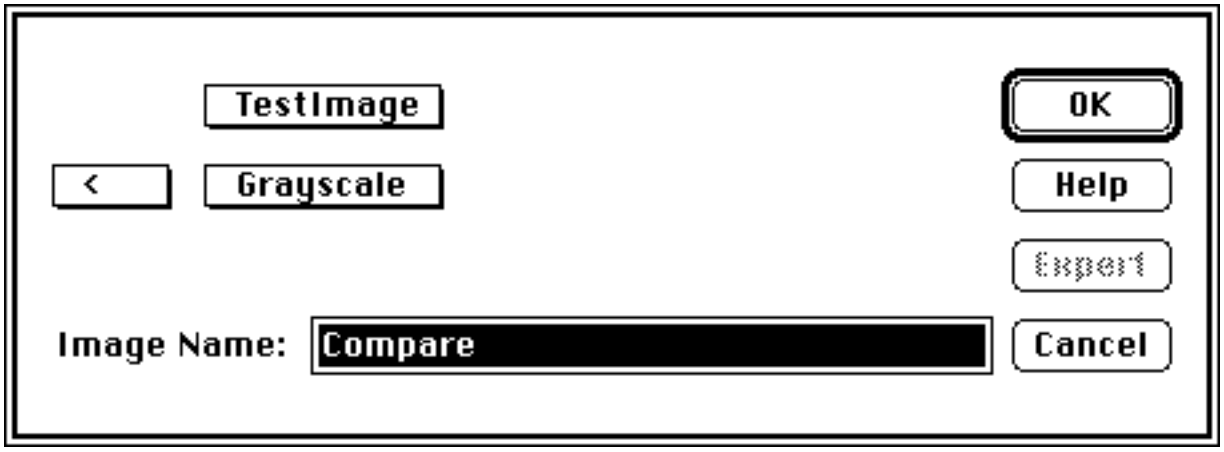
where w_I and w_J are the weights for the first and second image, respectively, as specified in the dialog. Note that if you want to compute the quotient of images I and J, w_J must not be equal to zero; if J_{ijk} equals zero, then O_{ijk} is set to O_{ijk-1} (or, if J_{ijk} equals zero and k equals one, O_{ijk} is set to zero).

The Combine command is useful, for example, for detecting changes in images (combination function "–"), or for overlaying binary text or graphics onto an image (combination functions "min" and "max").

Compare



The Combine command computes a binary output image, representing a comparison between corresponding pixels in two input images. Parameters are the two input images and the comparison function. The first image defaults to the image in the frontmost window, the second image to the one directly behind it. The two images, which must have the same number of bands, lines/band, and pixels/line, are chosen from pop-up menus of all images currently on the desktop. The comparison function is chosen from a pop-up menu, with all menu choices permanently enabled. The input images are treated as binary images in the case of the logical connectives "and," "or," "xor," and "not;" with a graylevel value of zero being the equivalent of the boolean value "false," and all other graylevel values being equivalent to "true."



The graylevel of a pixel in the output image *O* is computed from the graylevels of the corresponding pixels in the input images *I* and *J* as

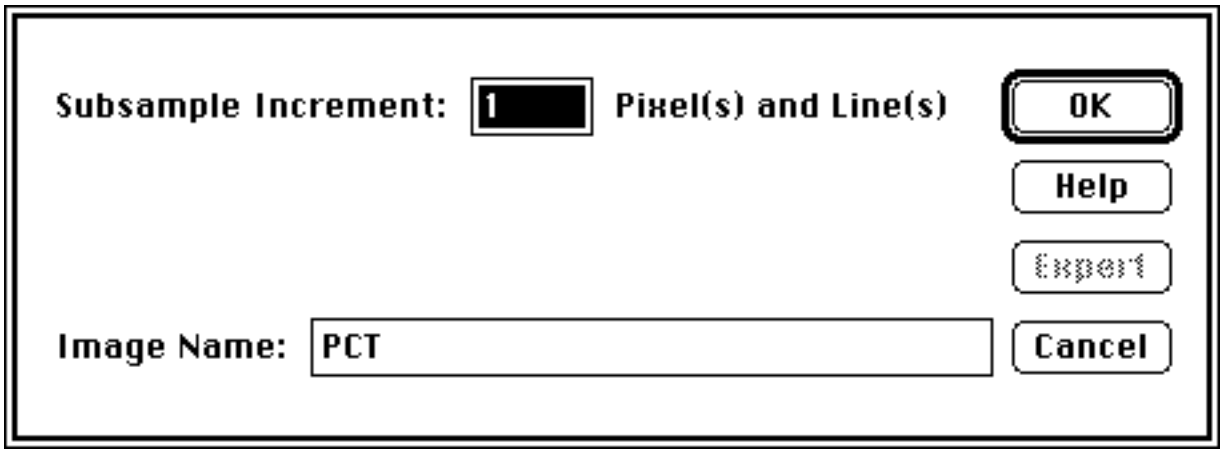
$$O_{ijk} = \begin{cases} \begin{cases} 1 & \text{if } I_{ijk} < J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if "<" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} \le J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if "≤" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} > J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if ">" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} \ge J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if "≥" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} = J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if "=" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} \ne J_{ijk} \\ 0 & \text{otherwise} \end{cases} & \text{(if "≠" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} \ne 0 \ \wedge \ J_{ijk} \ne 0 \\ 0 & \text{otherwise} \end{cases} & \text{(if "and" is specified)} \\ \begin{cases} 1 & \text{if } I_{ijk} \ne 0 \ \vee \ J_{ijk} \ne 0 \\ 0 & \text{otherwise} \end{cases} & \text{(if "or" is specified)} \\ \begin{cases} 1 & \text{if } \left(I_{ijk} \ne 0 \wedge J_{ijk} = 0\right) \vee \left(I_{ijk} = 0 \wedge J_{ijk} \ne 0\right) \\ 0 & \text{otherwise} \end{cases} & \text{(if "xor" is specified)} \\ \begin{cases} 1 & \text{if } J_{ijk} = 0 \\ 0 & \text{otherwise} \end{cases} & \text{(if "not" is specified)} \end{cases}$$

Note that the output image is a function of only the second input image in the case of the "not" combination function.

The Compare command is useful for creating and manipulating binary images and masks.

Principal Components Transform

The Principal Components Transform (PCT) command performs a PCT (also known as Karhunen-Loeve (KL) transform) on the input image. The subsample increment parameter lets you define the pixel and line increment for graylevel subsampling.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

$$O_{ijk} = \sum_{l=1}^{m} I_{ljk} \cdot E_{l_i}$$

where the input image is of size m-by-n-by-o, and E_l, $1 \le l \le m$, are the eigenvectors of the covariance matrix of the input image. The covariance matrix (see the description of the Analysis/Covariance command for a definition) is computed among the bands of the input image. The subsample increment parameter allows you a tradeoff between accuracy and speed in the calculation of the covariance matrix. A smaller value uses more pixels to compute the statistics and is therefore relatively slow; a larger value uses fewer pixels and is faster.

The Principal Components Transform command decorrelates the different bands of an image and orders the transformed bands by decreasing graylevel variance.

RGB → HSV

The RGB → HSV command transforms a color image from the RGB (red-green-blue) color space to the HSV (hue-saturation-value [or hue-saturation-brightness]) space. The transformation requires that all graylevel values in the input image be equal to or greater than zero.

The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

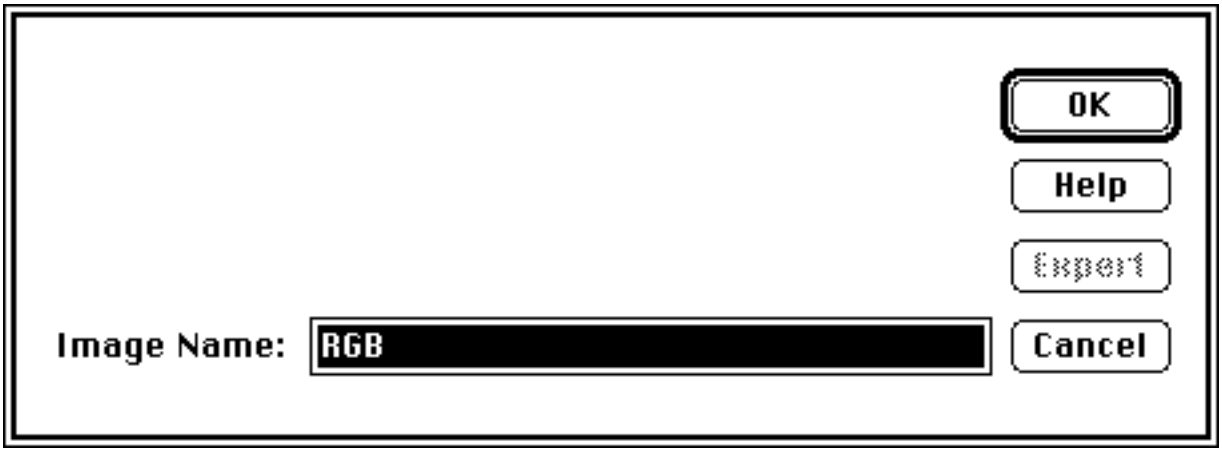
$$O_{ijk} = \text{hsv}(I_{ijk})$$

where hsv() is the color space transformation function. This function is described in detail in A. R. Smith, "Color Gamut Transform Pairs," *Computer Graphics*, Vol. 12, No. 3 (August 1978), pp. 12–19.

The RGB → HSV command is the functional inverse of the HSV → RGB command, which is described below.

HSV → RGB

The HSV → RGB command transforms a color image from the HSV (hue-saturation-value [or hue-saturation-brightness]) color space to the RGB (red-green-blue) space. The transformation requires that all graylevel values in the input image be in the range $[0.0, 255.0]$.



The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

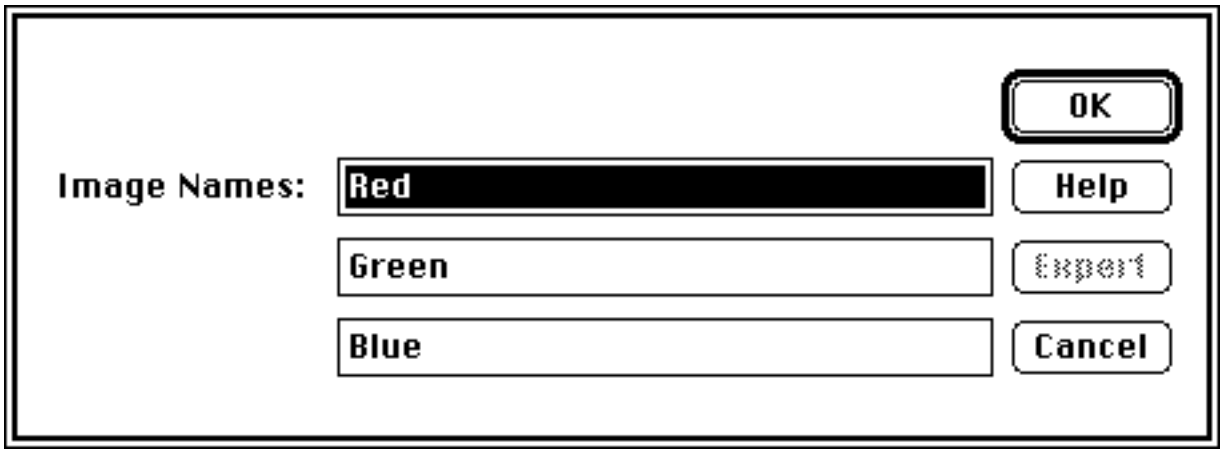
$$O_{ijk} = \text{rgb}(I_{ijk})$$

where rgb() is the color space transformation function. This function is described in detail in A. R. Smith, "Color Gamut Transform Pairs," *Computer Graphics*, Vol. 12, No. 3 (August 1978), pp. 12–19.

The HSV → RGB command is the functional inverse of the RGB → HSV command, which is described above.

BIL → BSQ

The BIL → BSQ (band-interleaved to band-sequential) command separates a three-band (color) image into three single-band images.



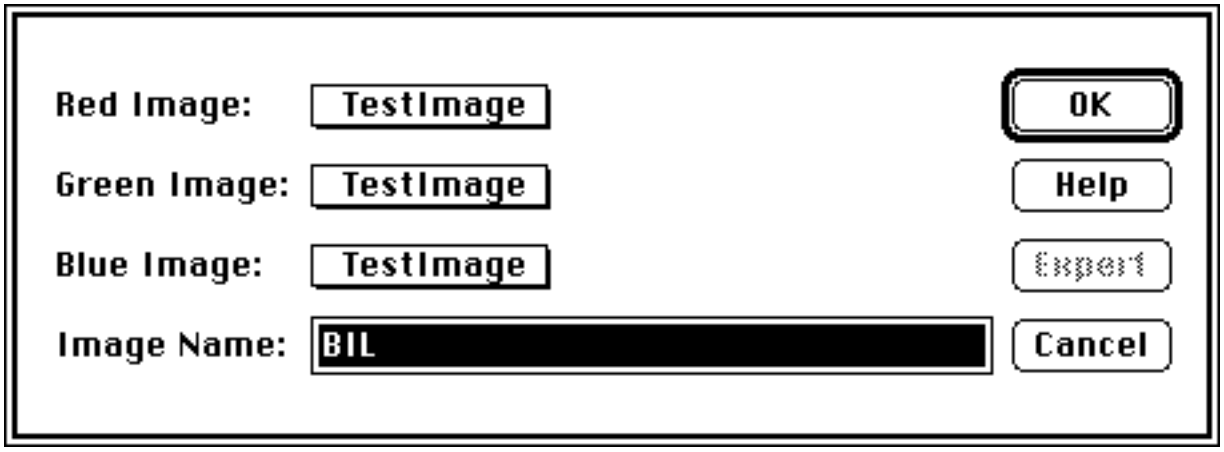
The graylevel of a pixel in the output image *R*, *G*, or *B* is computed from the graylevel of the corresponding pixel in the input image *I* as

$$\begin{aligned} R_{jk} &= I_{1jk} \\ G_{jk} &= I_{2jk} \\ B_{jk} &= I_{3jk} \end{aligned}$$

The BIL → BSQ command is useful for separating individual bands for further processing.

BSQ → BIL

The BSQ → BIL (band-sequential to band-interleaved) command combines three single-band images into a three-band (color) image. Parameters are the three input images, which all default to the image in the frontmost window. The three input images, which must all be single-band and have the same number of lines/band and pixels/line, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled.



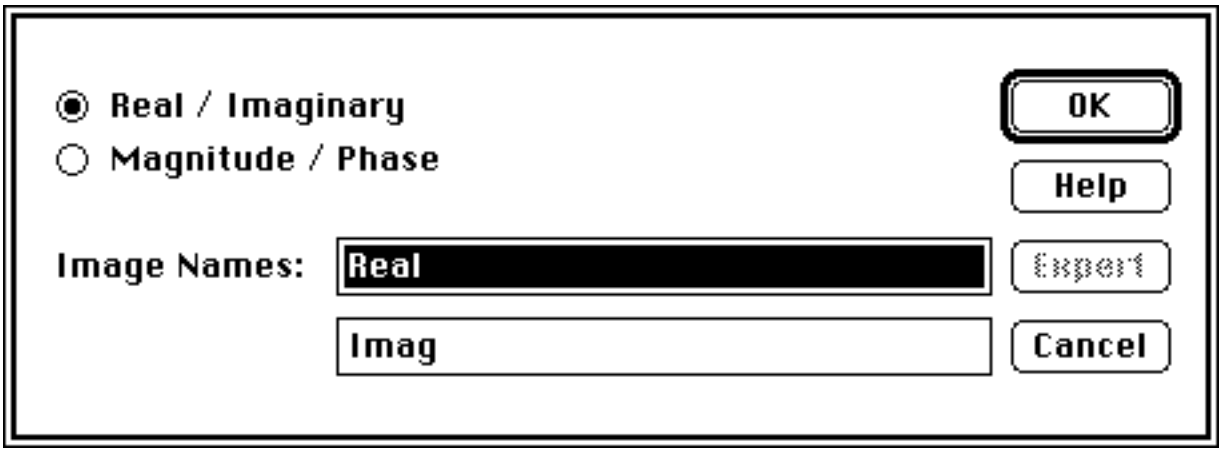
The graylevel of a pixel in the output image *O* is computed from the graylevels of the corresponding pixels in the input images *R*, *G*, and *B* as

$$O_{ijk} = \begin{cases} R_{jk} & \text{if } i = 1 \\ G_{jk} & \text{if } i = 2 \\ B_{jk} & \text{if } i = 3 \end{cases}$$

The BSQ → BIL command is useful for composing three images into a color composite. Note that the output image will appear to be a grayscale image if the three input images are identical; however, it is nonetheless a three-band image.

Complex → Real

The Complex → Real command converts an input image in complex format to two images in real format. You need to indicate whether the complex values should be separated into their real and imaginary or magnitude and phase components.



The graylevel of a pixel in the output images *O* and *P* are computed from the input image *I* as

$$O_{ijk} = \begin{cases} I_{ij\,2\cdot k} & \text{(if Real/Imaginary is specified)} \\ \sqrt{I^2_{ij\,2\cdot k} + I^2_{ij\,2\cdot k+1}} & \text{(if Magnitude/Phase is specified)} \end{cases}$$

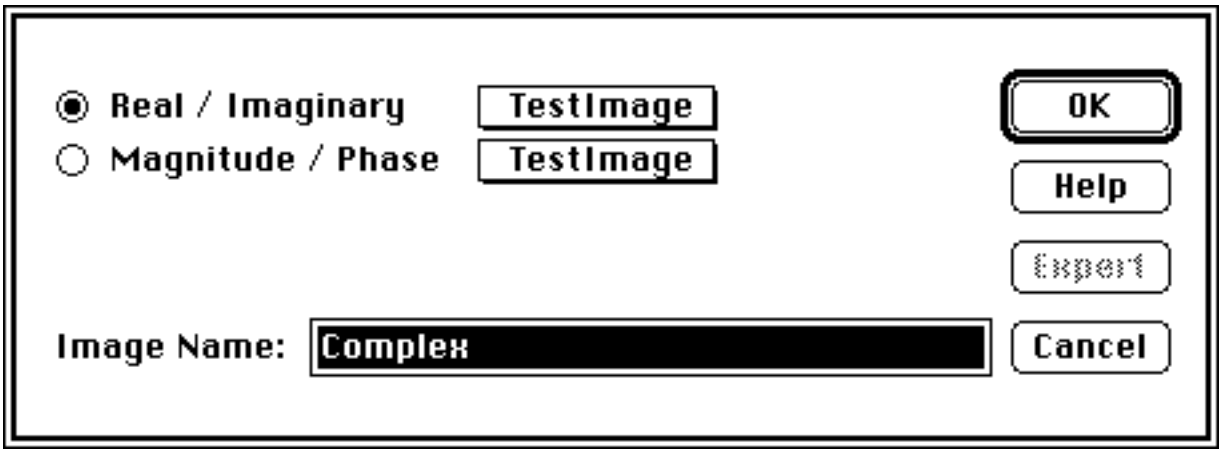
$$P_{ijk} = \begin{cases} I_{ij\,2\cdot k+1} & \text{(if Real/Imaginary is specified)} \\ \tan^{-1}\left(\dfrac{I^2_{ij\,2\cdot k+1}}{I^2_{ij\,2\cdot k}}\right) & \text{(if Magnitude/Phase is specified)} \end{cases}$$

where $\tan^{-1}()$ represents the trigonometric arc tangent function.

The Complex → Real command is useful for separating an image in complex format created by the Filter/FFT or Filter/FHT command into its components (in real format) for further processing.

Real → Complex

The Real → Complex command converts two input images in real format to one image in complex format. You need to indicate whether the input images represent the real and imaginary or magnitude and phase components of the complex image. Both images default to the image in the frontmost window. The two input images, which must have the same number of bands, lines/band, and pixels/line, are chosen from pop-up menus of all images currently on the desktop, with all menu choices permanently enabled.



The graylevel of a pixel in the output image O is computed from the input images I and J as

$$O_{ijk} = \begin{cases} I_{ij(k+1)/2} & \text{if } k \text{ is odd} \\ J_{ijk/2} & \text{if } k \text{ is even} \end{cases} \quad \text{(if Real/Imaginary is specified)}$$

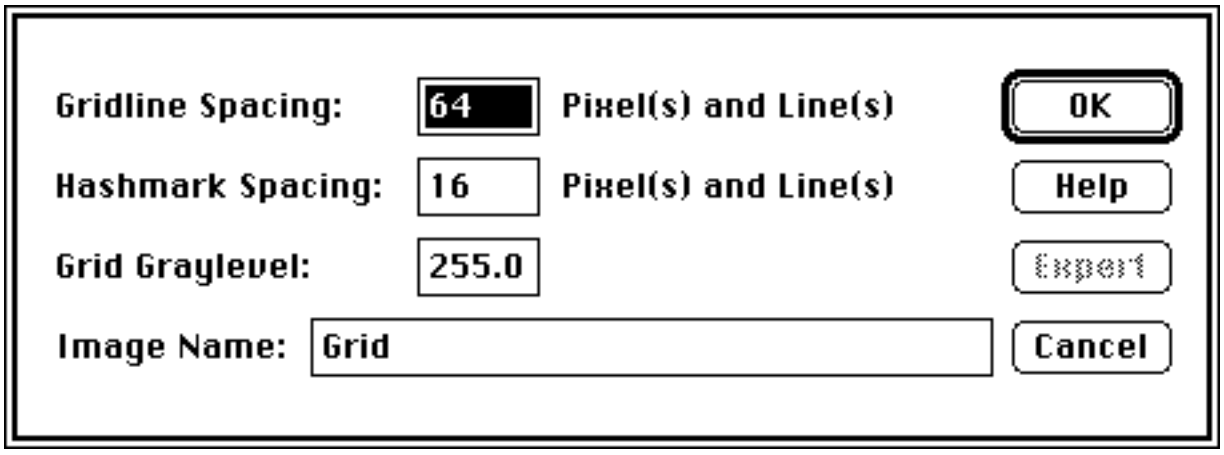
$$O_{ijk} = \begin{cases} I_{ij(k+1)/2} + \cos\left(J_{ij(k+1)/2}\right) & \text{if } k \text{ is odd} \\ I_{ijk/2} + \sin\left(J_{ijk/2}\right) & \text{if } k \text{ is even} \end{cases} \quad \text{(if Magnitude/Phase is specified)}$$

where cos() and sin() represent the trigonometric cosine and sine functions, respectively.

The Real → Complex command is useful for creating an image in complex format from two images in real format, as required by the Filter/FFT and Filter/FHT commands.

Grid

The Grid command overlays the input image with a grid of lines and hashmarks. Parameters are the distances between the gridlines and the hashmarks, respectively, and the graylevel value to be used for pixels on the grid. The grid graylevel defaults to the input image's maximum graylevel, I_{max}.



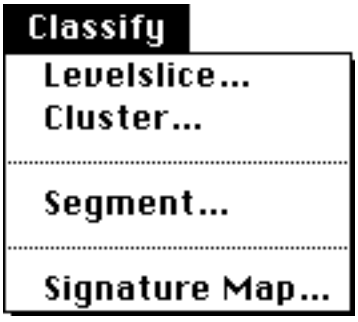
The graylevel of a pixel in the output image O is computed from the graylevel of the corresponding pixel in the input image I as

$$O_{ijk} = \begin{cases} g_{grid} & \text{if } \begin{array}{l} \operatorname{mod}(j-1, \Delta g) = 0 \ \vee \\ \operatorname{mod}(k-1, \Delta g) = 0 \ \vee \\ \operatorname{mod}(\operatorname{mod}(j-1, \Delta g), \Delta h) \le 2 \ \wedge \ \operatorname{mod}(\operatorname{mod}(k-1, \Delta g), \Delta h) = 0 \ \vee \\ \operatorname{mod}(\operatorname{mod}(j-1, \Delta g), \Delta h) \ge \Delta h - 2 \ \wedge \ \operatorname{mod}(\operatorname{mod}(k-1, \Delta g), \Delta h) = 0 \ \vee \\ \operatorname{mod}(\operatorname{mod}(k-1, \Delta g), \Delta h) \le 2 \ \wedge \ \operatorname{mod}(\operatorname{mod}(j-1, \Delta g), \Delta h) = 0 \ \vee \\ \operatorname{mod}(\operatorname{mod}(k-1, \Delta g), \Delta h) \ge \Delta h - 2 \ \wedge \ \operatorname{mod}(\operatorname{mod}(j-1, \Delta g), \Delta h) = 0 \end{array} \\ I_{ijk} & \text{otherwise} \end{cases}$$

where Δg is the gridline spacing, Δh is the hashmark spacing, and g_{grid} is the grid graylevel as specified in the dialog, and mod() represents the modulo function.

Note that, due to MacSADIE's automatic display stretch, the image contrast will be changed if the value of the grid graylevel is outside the graylevel range of the input image.

The Classify Menu

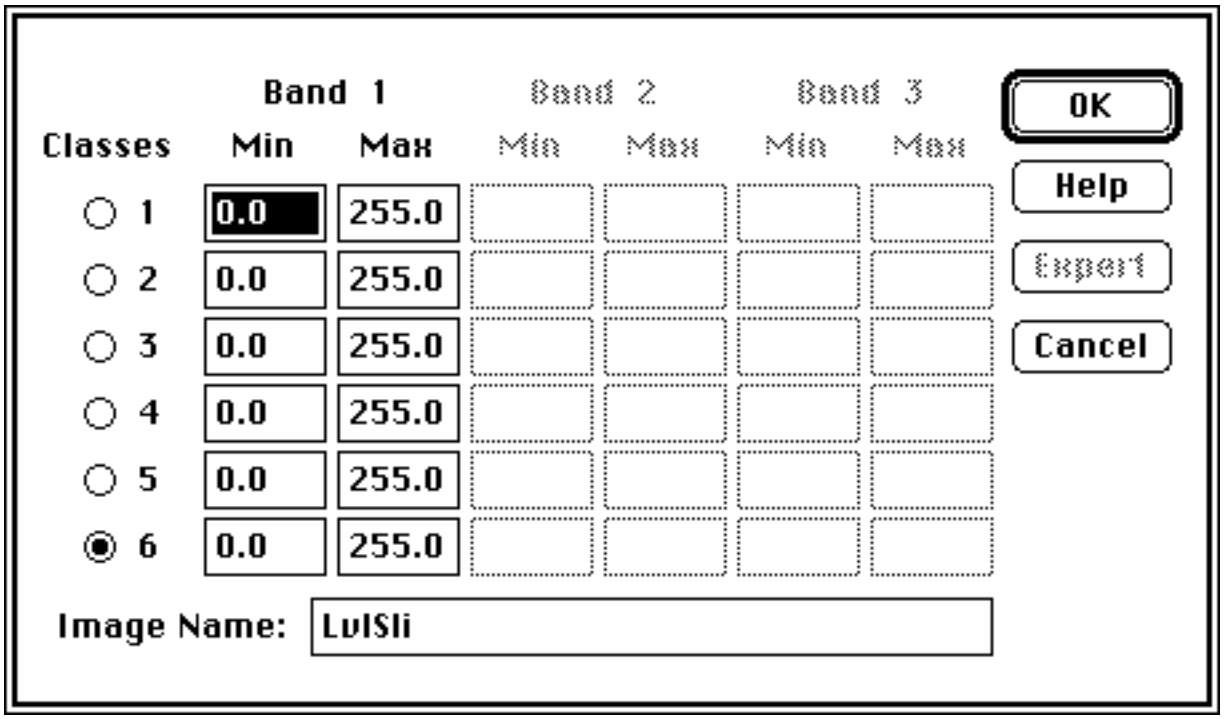


The Classify menu provides commands for applying supervised and unsupervised image classification (labeling) techniques to an image, as well as a command for creating a class signature map from a label map and the original image.

The Levelslice command is enabled for all images with no more than three bands. The Cluster and Segment commands are permanently enabled. The Signature Map command operates on two input images, and is therefore enabled only when there are at least two images on the desktop.

Levelslice

The Levelslice command classifies pixels in the input image, using a supervised level slice or "box" classifier. The command allows the specification of up to six classes. A class is defined by its graylevel minimum and maximum in each band of the image. The input image must not have more than three bands. Class boundaries are retained between invocations of the Levelslice command.



The label of a pixel in the output label map O is computed from the graylevel of the corresponding pixel in the input image I as

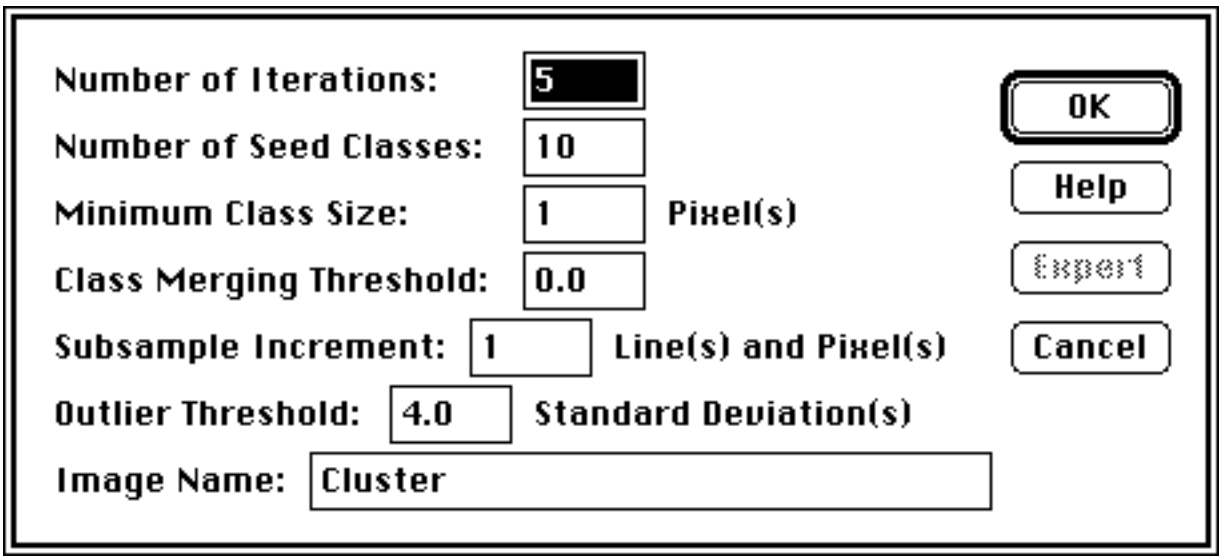
$$O_{jk} = \begin{cases} c & \text{if } \forall\, i,\ 1 \le i \le m\text{: } g_{c_i} \le I_{ijk} \le h_{c_i} \\ 0 & \text{otherwise} \end{cases}$$

where the input image is of size *m*-by-*n*-by-*o*, and g_{c_i} and h_{c_i} are the graylevel minimum and maximum, respectively, for class *c* and band *i*, as specified in the dialog. Pixels falling within two or more class boundaries are assigned labels by the nearest-mean algorithm.

The Levelslice command is useful for rapid, non-parametric supervised image classification.

Cluster

The Cluster command classifies pixels in the input image, based on their "natural grouping" in feature space. The classes (clusters) are determined in an iterative process. Parameters are the number of iterations, the number of seed classes, the minimum class size, a distance threshold for merging adjacent classes, the pixel and line increment for graylevel subsampling (used only in the iterative determination of the clusters, but not in the final classification), and an outlier threshold, which is used only in the final classification.



Pixels are assigned labels based on the proximity of their graylevel values to the graylevel means of the clusters. Initially, the class means are distributed uniformly across the *m*-dimensional feature space. Using the city-block distance criterion, each pixel is then assigned to the class with the closest mean. New class means are computed from those clusters, and the pixels are re-assigned to the classes. This process is repeated until the number of iterations specified in the dialog has been reached. If, during the iterative process, the number of pixels in any given class falls below the minimum class size specified in the dialog, that class is removed from subsequent iterations by allowing its pixels to be absorbed into other classes. Similarly, if two cluster means become, in each feature dimension, closer than the class merging threshold specified in the dialog, the corresponding classes are merged. Two classes *c* and *d* are merged if

$$\forall\ i,\ 1 \le i \le m\text{:}\ \frac{|\mu_{c_i} - \mu_{d_i}|}{(\sigma_{c_i} + \sigma_{d_i})} \le t_M$$

where μ_{c_i} and μ_{d_i} are the graylevel means for classes *c* and *d* in band *i*, σ_{c_i} and σ_{d_i} are the graylevel standard deviations for classes *c* and *d* in band *i*, and t_M is the class merging threshold, as specified in the dialog. Recommended values for this parameter are between 0.0 (no merging)

and 1.0. If two classes are merged, the resultant class is not permitted to be merged with any other class in the same iteration. If the minimum class size and the cluster merging threshold are set to one and zero, respectively, the final number of classes is equal to the number of seed classes. This type of clustering algorithm is described in more detail in R. D. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, J. Wiley & Sons, New York, NY, 1973.

In the final classification pass, the label of a pixel in the output label map *O* is computed from the graylevel of the corresponding pixel in the input image *I* as

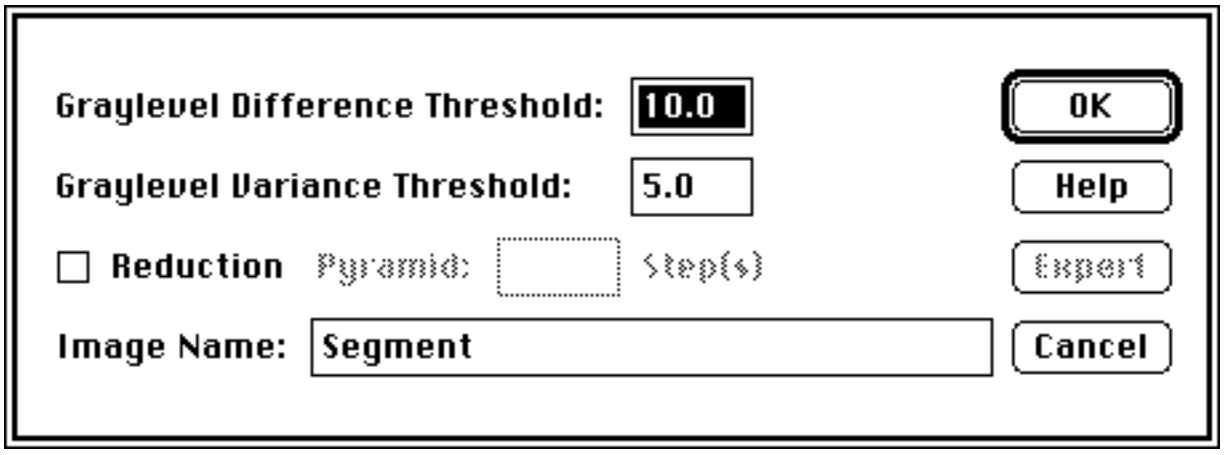
$$O_{jk} = \begin{cases} c & \text{if } \forall\, d,\ 1 \le d \le N\text{: } \sum_{i=1}^{m} |I_{ijk} - \mu_{c_i}| \le \sum_{i=1}^{m} |I_{ijk} - \mu_{d_i}| \ \wedge \\ & \qquad \forall\, i,\ 1 \le i \le m\text{: } \dfrac{|I_{ijk} - \mu_{c_i}|}{\sigma_{c_i}} \le t_O \\ 0 & \text{otherwise} \end{cases}$$

where the input image is of size *m*-by-*n*-by-*o*, *N* is the number of classes as determined by the iterative process, μ_{c_i} and μ_{d_i} are the graylevel means for classes *c* and *d* in band *i*, respectively, σ_{c_i} is the graylevel standard deviation for class *c* in band *i*, and t_O is the outlier threshold, as specified in the dialog.

The Cluster command is useful for non-parametric, unsupervised image classification. The stability of the clusters is determined, to a large extent, by the number of iterations. Using this algorithm, image data generally converges to a stable result after three or four iterations; however, it is advisable to look at the average mean vector migration values to confirm stability. The subsample increment parameter, which is used only in the iterative determination of the clusters, but not in the final classification, allows a tradeoff between accuracy and speed. A smaller value uses more pixels to determine the clusters and is therefore relatively slow; a larger value uses fewer pixels and is faster. When image subsampling is used, an increase in the average mean vector migration value is common for the final classification. This is usually a consequence of the change to full sampling and does not necessarily represent a true increase in average migration. Note that if subsampling is used, it may be advisable to increase the minimum class size.

Segment

The Segment command classifies pixels in the input image based on graylevel homogeneity and region connectivity. Parameters are two graylevel thresholds, which together define the homogeneity of the resulting regions. The spatial resolution of the input image may optionally be reduced by pixel averaging and subsampling before the image is subjected to the segmentation process. In this case the resulting label map is refined back to the resolution of the original input image. Resolution reduction is done in steps, with one step (the default, when reduction is chosen) being equal to a change in resolution by a factor of two.



Pixels in the output image are processed from left to right and top to bottom. If a pixel can be merged (according to the criterion given below) with either the region directly above or directly to the left of it, it is assigned that region's label. If it can be merged with either segment, it is assigned the label of the "closer" segment (again in the sense of the merge criterion). Otherwise, it begins a new region and is assigned a new label. If the pixel is merged with one of the two neighboring segments, a test is performed to see if those two segments can be merged. This technique ensures that all created regions are four-connected.

Two regions *P* and *Q* are merged if

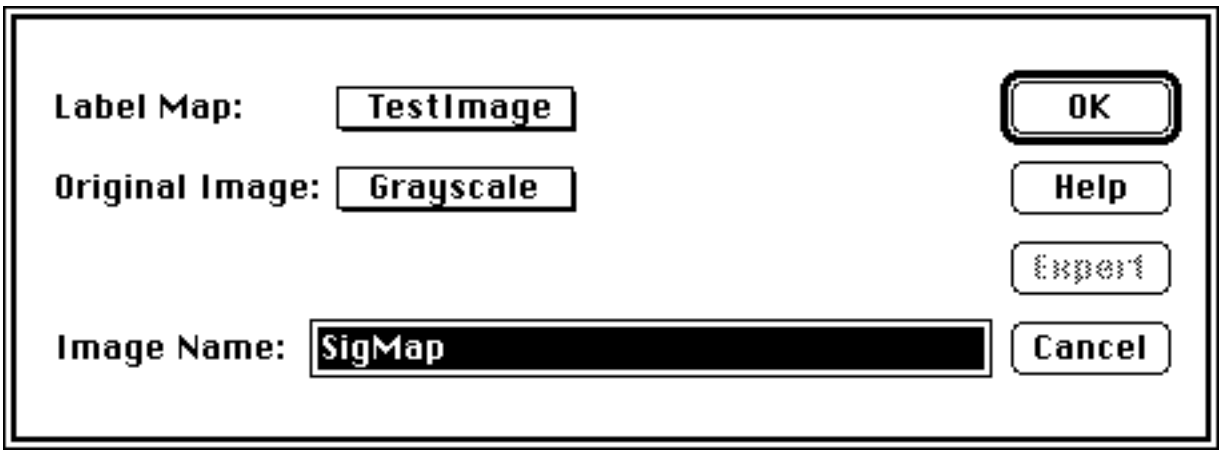
$$\forall\ i,\ 1 \le i \le m\text{:}\ |\mu_{P_i} - \mu_{Q_i}| \ \le\ t \cdot \left(1 - \frac{\sigma_{PQ_i}}{s}\right)$$

where the input image is of size *m*-by-*n*-by-*o*, *t* is the graylevel difference threshold, and *s* is the graylevel variance threshold, as specified in the dialog, μ_{P_i} and μ_{Q_i} are the mean graylevel values for regions *P* and *Q* in band *i*, respectively, and σ_{PQ} is the graylevel standard deviation for the resulting (merged) region in band *i*. The details of the segmentation algorithm, as well as the resampling and refining steps, are described in G. Mehldau, "An Image Segmentation Algorithm," University of Arizona, Digital Image Analysis Lab, Technical Report No. 90–3 (November 1990). The segmentation procedure is a modification of the technique described in T. Ryan, "Image Segmentation Algorithms," *SPIE*, Vol. 534 (1985), pp. 172–178.

The Segment command is useful for unsupervised image classification of spatially homogeneous regions. Each such region has a unique label (class) assignment. The label map may therefore be useful for counting objects on a uniform background. When the label map is combined with the original image via the Signature Map command (see below), the result is as a non-linearly smoothed version of the original image. This smoothed image may then be submitted to the Levelslice or Cluster commands for feature space classification. Note that resolution reduction is useful to avoid too small or too many segments; however, it does not necessarily result in a reduction in computation time. Also note that segmenting an image is a memory-intensive operation.

Signature Map

The Signature Map command creates a class signature map from a label map and the original image. The two images, which must have the same number of lines/band and pixels/line, are chosen from pop-up menus of all images currently on the desktop, with all choices permanently enabled. The label map defaults to the image in the frontmost window, the original image to the one directly behind it.



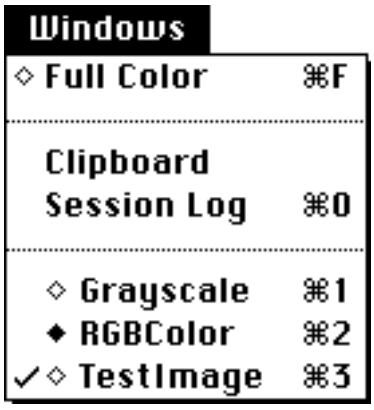
The graylevel of a pixel in the output image O is computed from the label of the corresponding pixel in the label map I and from the original image J as

$$O_{ijk} = \frac{1}{|S|} \cdot \sum_{J_{imn} \in S} J_{imn}$$

where $S = \{J_{imn} : I_{imn} = I_{ijk}\}$ (i.e., each pixel in the class signature map is represented by the average graylevel of its class).

The Signature Map command is useful for visualizing the performance of a classifier in terms of its effect on the image signatures. It is particularly useful for results produced by the Segment command, because its label map typically has a very large dynamic range and the labels are not correlated to the graylevels of the original image. The signature map may be subtracted from the original image (using the Misc/Combine command) to visualize the classification "reliability." Generally, the greater the deviation of a pixel from its classification signature (class mean), the lower the reliability. Note that because of MacSADIE's automatic display stretch, the displayed graylevels or colors of the signature map may not match those of the original image. They can be made consistent with the techniques described in the section on Grayscale and Color Display in the Introduction chapter of this manual.

The Windows Menu



The Windows menu provides a command for setting MacSADIE's color mode. In addition, it contains one entry for each MacSADIE window on the desktop.

The Full Color command is permanently enabled. The Clipboard and Session Log entries on the Windows menu are enabled only when the corresponding windows are visible. The entries for all other windows are always enabled.

Full Color

The Full Color command determines the color mode, and thus the appearance of grayscale and color images on the desktop, as described in the section on Grayscale and Color Display in the Introduction chapter of this manual. The Full Color command is a toggled command, with the diamond symbol in front of the command indicating the current mode. A hollow diamond indicates that Full Color mode is turned off; a filled diamond indicates that Full Color mode is turned on. The initial state of the Full Color command is dependent on the Monitors setting of the main screen at the time MacSADIE is started. With a setting of 256 colors (or less), Full Color mode is initially turned off; with a setting of more than 256 colors, Full Color mode is initially turned on.

Note that the color mode of an image is determined solely by the color mode in effect at the time the image was created — changing the color mode therefore does not affect existing images.

Clipboard

Selecting the Clipboard entry from the Windows menu makes the clipboard window the frontmost window on the desktop.

Note that the Clipboard entry on the Windows menu is available (enabled) only when the clipboard window is visible. You can change the visibility of the clipboard window with the Edit/Show Clipboard / Hide Clipboard command.

Session Log

Selecting the Session Log entry from the Windows menu makes the session log window the frontmost window on the desktop.

Note that the Session Log entry on the Windows menu is available (enabled) only when the session log window is visible. You can change the visibility of the session log window with the Edit/Hide Session Log / Show Session Log commands.

Window Entries

The remaining entries on the Windows menu show, from top to bottom, all image windows in the order of their creation. Selecting one of those entries makes the corresponding image window the frontmost window on the desktop.

The diamond symbol in front of a window name indicates the color mode of the corresponding image, as described under the Full Color command (above).

Appendix A: Mathematical Notation

X_{ijk} — Graylevel value of image X at band i, line j, and pixel k.
The first index (i) is omitted, if X is known to be a single-band image.

X_{min} , X_{max} — Minimum and maximum graylevel values of image X.

$f(x_1,x_2,\ldots)$ — Function f() of arguments x_1, x_2, ….
Examples: trunc(x) — the integer portion of a real number x
mod(x, y) — the remainder of the integer division x/y

$[x,y]$ — Interval from x to y, including the boundary values.
Examples: $[0, 255]$ — all integers from 0 to 255
$[0.0, 1.0]$ — all non-negative real numbers less than or equal to one

(x,y) — Interval from x to y, not including the boundary values.
Examples: $(0, 255)$ — all integers from 1 to 254
$(0.0, 1.0)$ — all positive real numbers less than one

Note: The interval notations above may be combined.
Example: $[0.0, 256.0)$ — all non-negative real numbers less than 256

$\lceil x \rceil$ — Ceiling of x. X, rounded up to the next integer.
Examples: $\lceil 1.2 \rceil = 2$, $\lceil 3.8 \rceil = 4$, $\lceil -1.2 \rceil = -1$

$\lfloor x \rfloor$ — Floor of x. X, rounded down to the next integer.
Examples: $\lfloor 1.2 \rfloor = 1$, $\lfloor 3.8 \rfloor = 3$, $\lfloor -1.2 \rfloor = -2$

$|x|$ — Size of x. If x is a number, the absolute value of x; if x is a set, the number of elements in x.
Examples: $|1.2| = 1.2$, $|-1.2| = 1.2$, $|\{1, 2, 3\}| = 3$

$\{x\}$ — Set containing x.
Example: $\{1, 2, 3\}$ — the set containing the numbers one, two, and three

$x \in y$ — Set membership.
Example: $I_{ijk} \in S$ — I_{ijk} is an element of set S

$x : y$ — All x for which y is true.
Example: $\{I_{ijk} : I_{ijk} \le x\}$ — the set of all I_{ijk} that are less than or equal to x

$\exists\, x : y$ — There exists an x such that y is true.
Example: $\exists\, i,j,k : I_{ijk} = 0$ — there exist indices i, j, k such that I_{ijk} equals zero

$\forall\, x : y$ — For all x, y is true.
Example: $\forall\, i,j,k :\ I_{ijk} < I_{ijk+1}$ — I_{ijk} is less than I_{ijk+1} for all indices i, j, k

$x \wedge y$ — Logical conjunction of x and y (true when both x and y are true; false otherwise).
Example: $I_{ijk} \le I_{lmn} \ \wedge\ I_{ijk} > I_{lmn}$ — false

$x \vee y$ — Logical disjunction of x and y (true when either x or y are true; false otherwise).
Example: $I_{ijk} \le I_{lmn} \ \vee\ I_{ijk} > I_{lmn}$ — true

$\neg x$ — Logical negation of x (true when x is false, false otherwise).
Example: $\neg(I_{ijk} \le I_{lmn} \ \vee\ I_{ijk} > I_{lmn})$ — false

Appendix B: Keyboard Shortcuts

Menus

It is possible to select frequently used MacSADIE commands directly from the keyboard by holding down the Command ( ) key and typing a single letter. The following is a complete list of those keyboard shortcuts, ordered by menu.

?	/ About MacSADIE
O	File / Open
W	File / Close
S	File / Save
P	File / Print
Q	File / Quit
Z	Edit / Undo
X	Edit / Cut
C	Edit / Copy
V	Edit / Paste
<delete>	Edit / Clear
A	Edit / Select All
I	Analysis / Image Information
H	Analysis / Statistics and Histogram
D	Analysis / RMS Difference
N	Contrast / Normalize Stretch
E	Contrast / Histogram Equalize Stretch
L	Contrast / Logarithmic Stretch
R	Contrast / Square Root Stretch
G	Filter / Gradient (Library Mask)
F	Windows / Full Color
0	Windows / Session Log
1 9	Windows / (oldest image) (newest image)

For most MacSADIE commands (with or without keyboard shortcuts), you can skip the dialog by holding down the Shift key when selecting the command via mouse or keyboard. Skipping the dialog causes the command to be executed with its default parameters. The exceptions to this rule are the /About MacSADIE, File/Open, File/Import, File/SaveAs, File/Page Setup, Edit/Preferences, and Analysis/Image Information commands, where this feature would defeat the purpose of the dialog, as well as all commands from the Edit and Windows menu that do not have a dialog. Holding down the Shift key when selecting the File/Quit command adds statistical output to the session log, and brings up the Print command's dialog for the session log.

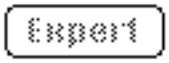
Dialogs

Most buttons in MacSADIE's dialogs have Command key equivalents, as shown in the following list.

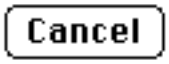
<return>
<enter>

H
?

E
!



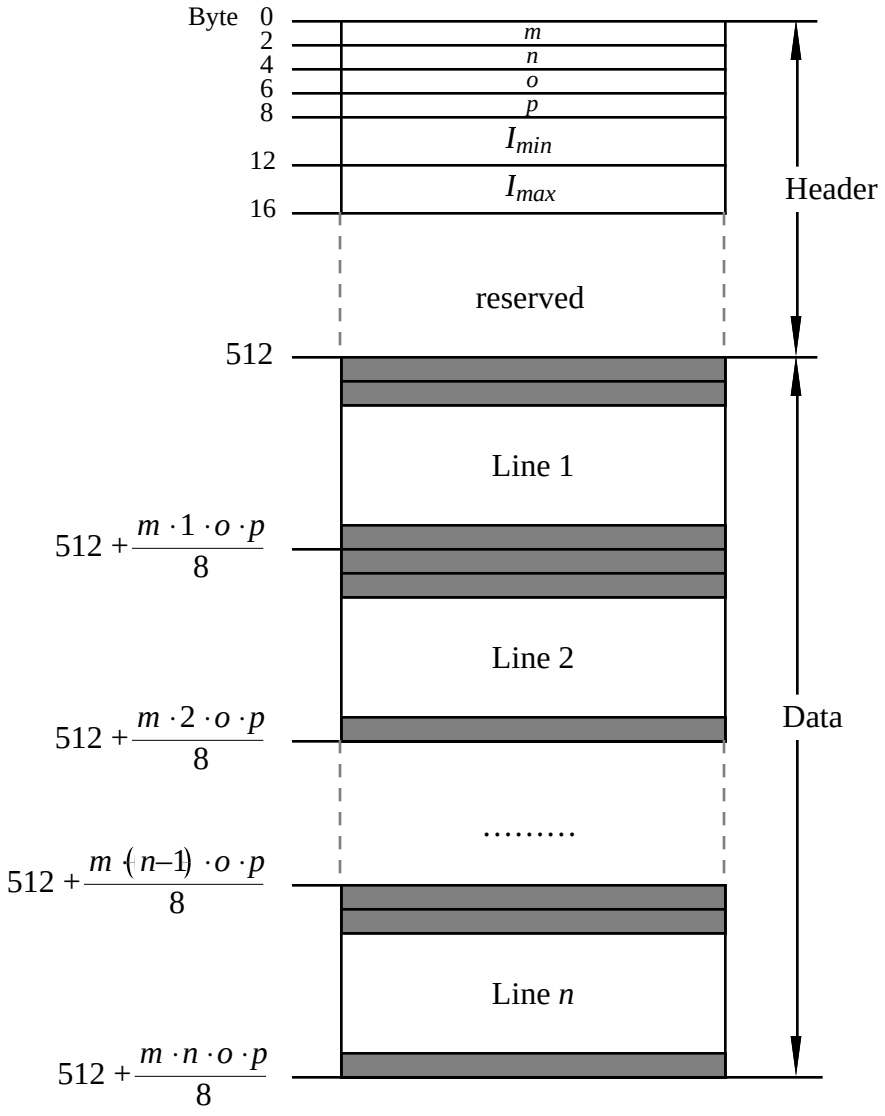
C
.



The File/Open, File/Import, and File/Save commands have a number of additional keyboard shortcuts, which are identical between applications. The keyboard shortcuts for the Cancel button do not work with the Page Setup and Print commands.

Appendix C: MacSADIE Image File Format

The following diagram shows the MacSADIE file format, assuming an image size of *m* bands, *n* lines, and *o* pixels, stored at *p* bits/pixel.



where is one line (*o* pixels) of image data from Band 1,

is one line (*o* pixels) of image data from Band 2,

and

is one line (*o* pixels) of image data from Band *m*.

Appendix D: MacSADIE Problem Report

Hardware Configuration

CPU (Model):
Memory (RAM):
Monitor (Make and Model):
Color Board (Make and Model):

Software Configuration

System Software:
System Version (from About the Finder…):
Finder Version (from About the Finder…):
MultiFinder Version (from About MultiFinder…):
32-Bit QuickDraw Version (from Get Info…):

MacSADIE Version (from About MacSADIE…):

INITs:

Problem Description

Briefly describe the problem you are experiencing. The description should include a specification of the data being used, the exact sequence of commands issued (together with their parameters), and a description of the obtained versus the expected results.

Mail To

MacSADIE c/o Digital Image Analysis Lab
Dept. of Electrical and Computer Engineering
University of Arizona
TUCSON, AZ 85721, U.S.A.
macsadie@dialsun.ece.arizona.edu

Index

A

About MacSADIE 17
Adaptive Stretch 50
Analysis Menu 33
Average and Subsample 70

B

Background Processing 15
Band-Interleaved Image 25, 86
Band-Sequential Image 25, 86
Bi-Linear Interpolation 69, 78
BIL
  Image 25, 86
  → BSQ 86
BSQ
  Image 25, 86
  → BIL 86
Button
  Cancel 14, 15, 101
  Done 33, 101
  Expert 14, 101
  Help 14, 101
  OK 14, 101
  Plot 34
  Window/Subsample 26
  Zoom In 34
  Zoom Out 34

C

Cancel Button 14, 15, 101
Checkerboard 19
Classification
  Supervised 91, 92
  Unsupervised 91, 93, 94
Classify Menu 91
Clear 13, 32
Clipboard 97
  Hide 32
  Show 32
Close 26
Cluster 92
Color 11
  Display 11
  Image 11
  Mode 13, 97
Combine 81
Command
  Disabled 14, 19, 31, 33, 41, 53, 69, 81, 91, 97
  Enabled 14, 19, 31, 33, 41, 53, 69, 81, 91, 97
  Key 11, 14, 15, 27, 100, 101
Compare 82
Complex
  Image 56, 57, 87, 88
  → Real 87
Conditional Average Filter 66
Contour Map 35
Contrast Menu 41
Control Panel 13
Control Points 75
Convolution 57
  FFT 53
  Wrap-Around 54, 55
Copy 13, 31
Correlation Matrix 36
Covariance 36
  Matrix 36
Covariance Matrix 84
Cut 13, 31

D

Dialog 14, 101
Display Stretch 11, 20, 42, 44, 49, 72, 73, 74, 89, 95
Diversity Map 62
Done Button 33, 101

E

Edit Menu 31
Error
  Code 15
  Input-Related 15
  Memory-Related 16
  Message 15
Expert Button 14, 101

F

FFT 55
  Convolution 53
  Filter 54
  Inverse Transform 55
FHT 56
  Inverse Transform 56
File Format 102
File Menu 19
Filter
  Conditional Average 66
  FFT 54
  High-Pass 54
  Low-Pass 54

Majority 63
Maximum 65
Median 64
Minimum 65
Parametric High-Boost 54, 55
Filter Menu 53
Fourier Transform 55
Frequency Domain Image 56, 57
Full Color 13, 97
Function 21

G

General Mosaic 74
Generic Image 24
Geometry Menu 69
Gradient
Library Mask 59
User Mask 60
Graylevels
Requantize 47
Saturate 48
Threshold 49
Grayscale 11, 20
Display 11
Image 11
Grid 88

H

Hardware Requirements 7
Hartley Transform 56
Help Button 14, 101
Hide
Clipboard 32
Session Log 32
High-Pass Filter 54
Histogram 34
Histogram Equalize Stretch 42
Horizontal Mosaic 71
HSV 84, 85
→ RGB 85
Hue-Saturation-Value 84, 85

I

Image 9
BIL 25, 86
BSQ 25, 86
Color 11
Complex 56, 57, 87, 88
File Format 102
Frequency Domain 56, 57
Generic 24
Grayscale 11
MacSADIE 8, 9, 24, 27, 28, 102
Multi-Band 11
PICT 8, 9, 24
Single-Band 11
Spatial Kernel 54, 57, 58
Synthetic 19
Text 32
TIFF 8, 9, 24
Image Information 33
Import 24
Insert 75
Installation 8
Interpolation
Bi-Linear 69, 78
Nearest Neighbor 69, 78
Parametric Cubic 69, 78
Inverse Transform
FFT 55
FHT 56

K

Karhunen-Loeve Transform 83
Key
Command 11, 14, 15, 27, 100, 101
Shift 14, 24, 25, 28, 29, 34, 100
Keyboard Shortcuts 100

L

Label Map 91, 93, 95
Levelslice 91
Linear Scale 69
Linear Stretch 43
Logarithmic Stretch 44
Low-Pass Filter 54

M

Macros 16
MacSADIE
About 17
Image
File Format 102
Open 8, 24
Print 9, 28
Save 27
Installation 8
Version Number 17
Majority Filter 63
Map
Diversity 62
Label 91, 93, 95
Variance 61
Mathematical Notation 99
Matrix

Correlation 36
Covariance 36, 84
Maximum Filter 65
Median Filter 64
Memory
Problems 16
Requirements 8, 10
Menu 14, 100
Analysis 33
Classify 91
Contrast 41
Edit 31
File 19
Filter 53
Geometry 69
Miscellaneous 81
Windows 97
 17
Minimum Filter 65
Mirror 79
Miscellaneous Menu 81
Monitors Setting 13
Mosaic
General 74
Horizontal 71
Vertical 72
Multi-Band Image 11

N

Nearest Neighbor Interpolation 69, 78
New 19
Normalize Stretch 41

O

OK Button 14, 101
Open 24
MacSADIE Image 8, 24
PICT Image 8, 24
TIFF Image 8, 24

P

Page Setup 28
Parametric Cubic Interpolation 69, 78
Parametric High-Boost Filter 54, 55
Paste 31
PICT Image
Open 8, 24
Print 9
Piecewise Linear Stretch 45
Plot Button 34
Polynomial Warp 77
Power Spectrum 37
Preferences 32
Principal Components Transform 83
Print 28
MacSADIE Image 9, 28
PICT Image 9
TIFF Image 9
Problem Report 103
Progress Indicator 15

Q

QuickDraw 8
Quit 29

R

Random 23
Real → Complex 88
Redo 31
Reference Stretch 50
Requantize Graylevels 47
Requirements
Hardware 7
Memory 8, 10
Software 8
RGB → HSV 84
RMS Difference 38
Rotate 79

S

Saturate Graylevels 48
Save 27
Save As 27
Scattergram 38
Segment 93
Select All 13, 32
Selection 13, 31, 32
Rectangle 13, 31, 32
Session Log 9, 97
Hide 32
Show 32
Shift Key 14, 24, 25, 28, 29, 34, 100
Shortcuts 100
Show
Clipboard 32
Session Log 32
Signature Map 95
Sinewave 21
Single-Band Image 11
Software Requirements 8
Spatial Kernel Image 54, 57, 58
Square Root Stretch 46
Square Stretch 46
Startup 8
Statistics 34

Stretch
Adaptive 50
Display 11, 20, 42, 44, 49, 72, 73, 74, 89, 95
Histogram Equalize 42
Linear 43
Logarithmic 44
Normalize 41
Piecewise Linear 45
Reference 50
Square 46
Square Root 46
Table Lookup 47
Subsample
Average 70
Import 26
Window 71
Supervised Classification 91, 92
Synthetic Image 19
System Software 8

T

Table Lookup Stretch 47
Text Image 32
Threshold Graylevels 49
TIFF Image
Open 8, 24
Print 9
Transform
Fourier 55
Hartley 56
Karhunen-Loeve 83
Principal Components 83

U

Undo 31
Unsupervised Classification 91, 93, 94

V

Variance Map 61
Version Number 17
Vertical Mosaic 72

W

Window 9, 97
Window and Subsample 71
Window Entries 97
Window/Subsample Button 26
Windows Menu 97
Wrap-Around Convolution 54, 55

Z

Zoom In Button 34
Zoom Out Button 34

 Menu 17